20. Library Administration

20.1 EMD Library Administration Overview

The EMD Library Administration is provided by several departments within the EMD project through the combined resources of Data Management (DM), Configuration Management (CM), and the System Monitoring Center (SMC). Library Administration includes (1) production, maintenance, and distribution of baselined EMD documents, (2) delivery of approved commercial off-the-shelf (COTS) software and documentation and non-contractual documentation to the DAACs and other Government facilities; and (3) DAAC specific production, maintenance, and distribution of documents which, due to a more timely need for document updates, are produced locally and are tailored to reflect individual DAAC needs and configurations. The DAAC specific updates are eventually incorporated into EMD approved documents. There are three web sites that function as electronic distribution points for the approved data and documents. These web sites are maintained by DM, CM and SMC respectively: The ECS Data Handling System (EDHS), the EMD Baseline Information System (EBIS), and the System Monitoring Center (SMC). All of these processes are discussed in more detail in this Library Administration section.

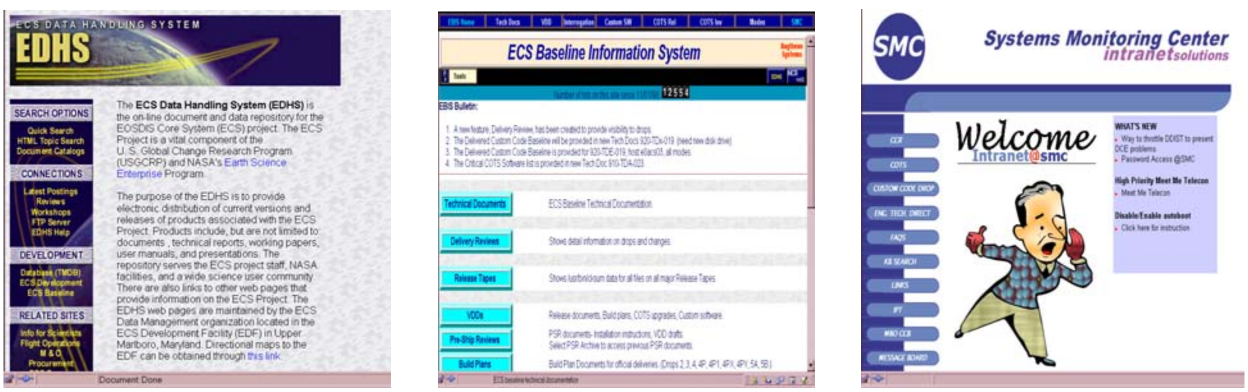

Figure 20-1. The EDHS Home Page, the EMD Baseline Information Home Page, and the System Monitoring Center Home Page

20.1.1 Data Management (DM)

DM is the focal point for establishing and advancing all project document/data management activities. In this regard, DM works closely with all EMD offices to provide efficient and cost-effective distribution, storage, maintenance, and retrieval of these data. DM is responsible for maintaining EMD data/documentation which includes documents under control of the EMD

Change Control Boards (CCB) and subordinate CCBs. DM's responsibilities encompass three functional areas: (1) Data Requirements refers to the deliverable documentation specified in the Contract Data Requirements List (CDRL) as well as other data items that document the EMD Project; (2) Data Control activities focus on the efficient archive, storage and maintenance of materials that support such things as milestone reviews, technical papers and white papers; and other pertinent data such as contract correspondence, progress reports, and background information; and (3) Data Support includes the preparation of documentation for publication. This includes format editing, document coordination, graphics, layout, and reproduction. The Document Coordination staff is responsible for all activities required to prepare CDRLs and other documentation required by the contract. To make documentation readily available, DM has established an electronic distribution via the World Wide Web through the EDHS, (http://edhs1.gsfc.nasa.gov).

20.1.1.1 Authoring Documents

EMD CDRLs and other documents are authored by project personnel using existing tools and templates to ensure consistency and completeness with customer requirements. A standard set of software applications are used across the EMD Project. The use of this common set of production tools by both the development personnel and the documentation staff reduces redundant activities such as keystrokes and art preparation.

20.1.1.2 Formatting Documents

To ensure compliance with customer standards and to promote consistency and ease of use, a standard tool kit of document formats or templates was developed by DM. These templates are located on the Internal Server portion EDHS web site under the Data Management Document Templates and Guidelines, and are used by authors to develop CDRLs and other types of documents. After a document is written or updated by an author, it is then forwarded to DM for further processing. DM assigns the document a unique document number and reviews the document for completeness and format accuracy.

20.1.1.3 Posting and Retrieval of Documents

After documents are formatted and reviewed by DM, they are reviewed and approved by the appropriate Configuration Change Board (CCB) and other reviewers as required. Approved documents, which are not otherwise restricted, are posted to the EDHS. The EDHS web site provides on-line search and retrieval of EMD documentation and is the primary repository of information maintained by the EMD Project. DM maintains the EDHS web pages and is responsible for the integrity of all posted documentation.

20.1.1.4 Distribution and Maintenance of Documents

EMD CDRLS and Required Documents are maintained by DM, for the life of the project. A Baseline change to an Earth Observing System Data and Information System (EOSDIS) approved document is accomplished through a document change notice (DCN) or revision. Documentation produced by the project is distributed internally and/or to the customer.

Dissemination includes printed hardcopy and/or electronic posting as indicated in the preceding section.

20.2 Configuration Management (CM) Overview

The EMD CM Office requirements and objectives in support of EMD Library Administration are to maintain and publish EMD Technical Baseline Documentation on the EMD Baseline Information System (EBIS), located on the following url: (http://cmdm.east.hitc.com/baseline/) or (http://pete.hitc.com).

20.2.1 Configuration Management (CM)

The EMD Technical Baseline documents are updated when the EMD CCB approve CCRs, which pertain to the DAACs. This naming convention is defined in 910-TDA-001, Baseline Specification Document.

The EBIS contains different types of documents within the EMD Project, such as:

1. Technical Documents, posted as CCRs are approved by CCB
2. Delivery Reviews, shows detail information on drops and changes
3. Release Tapes, Shows /us/bincksum data for all files on all major Release Tapes.
4. VDDs, Release documents, Build Plans, COTS upgrades, Custom Software.
5. Pre-Ship Reviews, PSR documents – Installation Instructions, VDD drafts.
6. Build Plans
7. Test Executables
8. EMD Configuration
9. Operational Modes
10. COTS Status
11. SMC WWWBoard

20.2.1.1 Posting/Retrieval of Documents/Software from Configuration Management Server

Information being disseminated by http://cmdm.east.hitc.com/baseline/) must have been approved by the CCB.

Documents are posted in PDF format, and are posted in accordance with 905-TDA-001. Documents are posted on the EBIS with the current revision level for that document. If there is any question relating to any document on the EBIS page, contact the CM Administrator listed on the WEB Site page.

20.2.1.2 EMD Software Library Maintenance

The EMD Software Library is responsible for controlling and tracking all approved COTS software for the project. CD's, tar files, and other media is disseminated to the SMC or DAACs depending on the type of COTS software. The Software Library maintains previous versions of COTS products and has the responsibility to ensure that only COTS products have an approved CCR before release of the product.

20.3 System Monitoring Center (SMC) System Overview

The SMC system requirements and objectives in support of EMD Library Administration are overall system performance monitoring, coordinating, and setting system wide polices and priorities

20.3.1 System Monitoring Center (SMC) System

The SMC servers are the distribution points for:

1. Staging area and distribution for EMD Custom/COTS Software deliveries.
2. Medium for the distribution of non-contractual documentation to the sites. Documents such as README file, COTS electronic instructions, Technical white papers, CCRs, NCR Workaround instructions, database scripts repository etc.…
3. Maintains copies of all deliverables that include, but are not limited to, binaries, executables, Toolkit deliveries, test data, NCR workarounds, README files, general instructions, etc.

Information can be retrieved from the SMC by accessing the Website. The URL is as follows (http://m0mss01.ecs.nasa.gov/smc/)

20.3.1.1 Posting/Retrieval of Documents/Software from System Monitoring Center (SMC) Server

The Systems Monitoring Center Staff will be responsible for the dissemination of information provide to the SMC. Information can come from a variety of sources such as EDF, DAACs, and others. The submitter will notify the SMC staff (via phone, pager, or e-mail) that the files are ready. The software and the supporting documentation can either be pushed to the sites or pulled by the sites for installation

20.3.1.2 Authoring Documents

A Systems Operation Support (EMD) sustaining engineer will create all source material (text, graphics files, etc.) per CDRL/DID preparation instructions and be accountable for the accuracy of its content. Local DAAC production personnel will assist the author by providing word-processing and graphics support such as templates and fonts.

20.3.1.3 Formatting Documents

All documents submitted to the EMD Book Boss by an EMD Engineer staff member should be in Microsoft Word format. These DAAC specific documents are numbered and controlled locally to include a review for completeness and format accuracy.

20.3.1.4 Submitting Documents

All documents submitted to the EMD Book Boss should be in soft copy. Soft copy should be sent via electronic mail as an attachment.

20.3.1.5 Requesting Documents

Requests for documents may be made by telephone, in person, or by electronic mail.

20.3.1.6 Metadata Maintenance

See Chapter 10, Metadata Administration.

20.4 On-Site Documentation Overview

On-site documentation requirements and objectives in support of EMD Library Administration are to generate site-specific documentation in accordance with program standards and conventions for format storage and control. EMD resources will be kept informed and utilized as appropriate

20.4.1 On-Site (DAAC-Specific) Baselined (CDRL/DID) Document Production, Maintenance, and Distribution

The on-site documentation activities are described in the paragraphs below.

20.4.1.1 Authoring Documents

The assigned engineer will create all source material (text, graphics files, etc.) per CDRL/DID preparation instructions & be accountable for the accuracy of its content. The DAAC's Book Boss will oversee the development of the documents and act as the interface between Local DAAC production personnel and the DAAC's assigned engineers. Local DAAC production personnel will assist the author by providing word-processing and graphics support such as templates and fonts.

The DAAC Book Boss can request the assistance/assignment of an EMD Book Boss to assist in the development of a document. In this case, the EMD Book Boss can serve as the focal point for editing, publication and dissemination of the document. The EMD Book Boss can also draw on EMD resources to assist in writing sections as needed or requested by the DAAC Book Boss.

20.4.1.2 Submitting and Formatting Documents

All documents submitted to the DAAC's Book Boss should be in MS Word format and also, if possible, in hard copy. Soft copy should be sent via electronic mail as an attachment and hard copy could be hand delivered to EMD Library. These DAAC specific documents are numbered and controlled locally to include a review for completeness and format accuracy.

20.4.1.3 Requesting Documents

Requests for documents may be made by telephone, in person, or by electronic mail.

20.4.1.4 Metadata Maintenance

Science Coordinator technically interfaces with the Science Computing Facilities by updating Interface Control Documents (ICD).

20.4.1.5 Maintenance of Baselined EMD Documents in Baseline Manager Tool

The Clearcase Baseline Manager Tool will be used to record the change history and updates to post current revisions to the Baselined EMD documents. This tool will also be used to provide the Master Index for the EMD Library.

20.4.1.6 Document Metadata Insertion Subscription

The EMD Library is the repository for all EMD Maintained documents (both hard copy and electronic). COTS documentation will be physically located in the library in it's own section. Documentation available on CD-ROM will be located in a separate cabinet. Documentation available on-line will be posted in the EMD Library and on the Document Management Server.

20.5 Systems Operation Support (EMD) COTS Overview

The EMD system requirements and objectives in support of EMD Library Administration are to enhance and streamline documentation activities related to operations and DAAC specific activities.

20.5.1 EMD COTS Document & Software Maintenance

The activities associated with documentation and software maintenance are described in the sections below.

20.5.1.1 COTS Library Database

The COTS library database designed in Microsoft Access is a means of controlling and maintaining Commercial Off the Shelf (COTS) documentation and software residing in the library. EMD personnel have access to the library database through the Document Management Server (DMS). The database enables EMD personnel to locate and retrieve document information. By using any of the available fields in the database, the user has the ability to search the contents of it. The library database search provides the user adequate information to retrieve the most current version of a document. Requesting COTS documentation may also be made via the database in which the EMD Librarian will receive notice as soon as the request is sent.

20.5.1.2 Document Access Control

The Libraries database enables EMD personnel to locate and retrieve document information. Only the librarian has the capability to add, edit, and delete. EMD personnel will have access to this database for visualization purposes via the Document Management Server.

20.5.1.3 COTS Library Reports

COTS Library inventory reports are generated as requested.

20.5.1.4 COTS Hardware and Software Procured by EMD to be Deployed for Use on the EMD Local Area Network

Each DAAC has a non-baselined hardware and software configuration, which supports their site office automation environment for non-production activities. Maintenance and Operations Hardware used by the DAAC's and SMC staff to monitor, analyze, report, and manage the operational baseline. EMD provides hardware and software for each site as specified in the DID 607 & 608 EMD documents. The EMD environment is not part of the CM controlled EMD baseline. COTS software installed on the production network will not be maintained in the EMD COTS library but, rather, by the DAAC CM staff, who will assure that they have copies of all software that is installed on the EMD network.

Hardware or software procured for a site to be installed on the production network will be received at the EDF and controlled under the procedures outlined in the EMD Property Management Plan and associated Project Instructions. After property is properly controlled it will be shipped to the appropriate site with a CCR showing how and where it should be installed).

20.5.1.5 On-Site COTS Document & Software Maintenance

EMD products deployed to the operational sites that have been released for operational use are maintained in the EMD Documentation and Software COTS Library maintained at each site (On-Site SW Library). Site personnel maintain partitioned libraries to facilitate access control of science software and other software not developed by EMD. Site personnel are responsible for any CM activities concerned with this library.

This page intentionally left blank.

21. COTS Hardware Maintenance

In this section, discussion of commercial-off-the-shelf (COTS) hardware maintenance support includes COTS hardware procured for the ECS Project and some Government furnished property (GFP).

The following documents are referenced in this section:

- Property Management Plan for the EMD Project 105-CD-001

21.1 COTS Hardware Maintenance - General

The Integrated Logistics Support (ILS) Office at the EDF is responsible for COTS hardware maintenance. Local Maintenance Coordinators (LMC) are responsible for implementation of EMD maintenance policy at the DAACs. Questions about COTS hardware maintenance policy are to be addressed to the ILS manager through the ILS Maintenance Coordinator, using the contact procedures found in the last sentence of this section.

COTS hardware maintenance consists of preventive and corrective maintenance. COTS hardware preventive maintenance is the responsibility of the contracted COTS hardware maintenance providers. COTS hardware corrective maintenance is the responsibility of the contracted COTS hardware maintenance providers, and/or the Local ILS Maintenance Coordinator (LMC) using local DAAC resources. The LMC is the DAAC's local point of contact for directing and coordinating corrective maintenance of EMD COTS hardware. COTS hardware maintenance support is available from contract COTS hardware maintenance providers according to the terms specified in each maintenance contract. Since all EMD hardware is either covered by an onsite, return to manufacturer, or time-and-material maintenance contract there is no need for robust site spares provided by the EMD program. Additionally EMD is an IDIQ Task Order contract, and Task 101, Sustaining Engineering, does not authorize purchasing additional spares; therefore, EMD will not purchase additional spares for Task 101. Depending on the task, additional site spares may be purchased for future tasks, but that decision will be made on a task-by-task basis. Any spares that remain from the ECS contract were transferred to the EMD contract and will continue to be used until exhausted. Site spares that may remain from ECS mainly consist of monitors, keyboards, mice, and a variety of circuit boards. To facilitate easy tracking of maintenance spares at each DAAC, the following method of identification will be used: in ILM, the ECSNAM for the spares will be the site name with "MAINT" added to the end [EDCMAINT, NSIDCMAINT, etc.]. The LMC ensures that all COTS hardware maintenance providers comply with requirements of this section and that accurate and timely information from the DAAC is entered in the Inventory-Logistics-Maintenance (ILM) System. **Instructions on the use of ILM are provided in chapter 27 of this manual.** The ILS Maintenance Coordinator is assigned to the ILS Office. The ILS Maintenance Coordinator is available 24x7 either by email or telephone during normal work hours or by pager for after-hour emergencies. The ILS Maintenance Coordinator's name, email address, phone, and pager numbers will be forwarded to the DAAC LMCs under separate cover.

21.1.1 Corrective Maintenance

Corrective maintenance is the unscheduled repair of equipment, to include fault detection, diagnosis, isolation, repair and recovery of software and databases if appropriate. The maintenance of hardware items may be performed on site by the LMC or the contracted maintenance provider, or by returning the failed component to the maintenance depot for repair or replacement.

21.1.2 Preventive Maintenance

Storage Technology automated tape library robots are currently the only hardware requiring scheduled preventive maintenance. Preventive maintenance is performed by the original equipment manufacturer (OEM) on this equipment. OEMs will coordinate preventive maintenance visits to the DAAC with the LMC. LMCs will record on the maintenance work orders (MWO) any downtime experienced as a result of preventive maintenance.

21.1.3 Configuration Management

Configuration Management (CM) requirements are addressed in Chapter 9 of this document. The LMC ensures compliance with the CM requirements resulting from a hardware maintenance action.

21.1.4 COTS Hardware Maintenance Safety

Hardware maintenance will be accomplished in a manner that ensures personnel and equipment are protected from harm. The LMC will ensue that local safety requirements are known and observed by local site support personnel and COTS hardware maintenance providers during COTS hardware maintenance.

COTS hardware safety practices include electrostatic discharge (ESD) protection. The ESD program will be locally developed by the LMC using applicable DAAC procedures for guidance. When not being worked on or when outside protected areas, electronic parts and assemblies are to be covered by ESD protective covering or packaging. During installation or removal of electronic parts or LRUs, a common ground will be established between the technician, work area, the part, and the equipment it is to be installed in/removed from. It is the responsibility of the LMC to ensure compliance with these safety procedures by the hardware maintenance provider and site personnel.

21.2 COTS Hardware Maintenance - Contract Information

The ECS COTS procurement organization is located at the ECS development facility (EDF) and is responsible for contracting for COTS hardware maintenance. Cost and support considerations may result in COTS HW maintenance being provided by a third-party provider. Questions or comments concerning COTS hardware maintenance are to be directed to the ILS Maintenance Coordinator.

21.2.1 COTS Hardware Maintenance Contract Database

Information relating to COTS hardware maintenance contracts is contained in the ILM system. The LMC can obtain specific information on all hardware maintenance contracts at the ILS web page, http://dmserver.gsfc.nasa.gov/ils/intro.htm. The ILS web page is kept current by the ILS Maintenance staff.

21.2.2 Information Required to Obtain COTS Hardware Maintenance

Generally, COTS hardware maintenance providers require an access, or site, code and/or the serial number of the host equipment to verify that the failed item is covered under a maintenance contract. For example, if maintenance were requested for a terminal/monitor or disk drive, the serial number of the parent workstation or server would need to be provided to the maintenance provider. The serial number may also be the access code for that provider. The information needed by the various COTS HW maintenance providers to verify that maintenance is authorized is specified on the ILS web page. DAAC site-specific site access numbers/site codes/contract numbers, if required, are also listed on the ILS web page. For some COTS HW maintenance providers, names of authorized contact persons are required. The number of authorized contact persons varies with the different maintenance providers. The ILS Maintenance Coordinator, in coordination with the LMC, arranges with the COTS HW maintenance provider for specified personnel to become authorized contact persons. It is the responsibility of the LMC to provide to the ILS Maintenance Coordinator the name changes to the authorized contact list as soon as known. The LMC will identify each change as a permanent or temporary change and, if temporary, the inclusive dates of the change. A temporary change may occur when the authorized contact person is ill, on vacation, in training, or other short-term change of work availability status has occurred or is expected to occur.

21.3 Hardware Repairs - Standard

Users/operators of ECS hardware should report hardware/system problems to the site's LMC for resolution. Users, operators, and support personnel who encounter a HW problem will report the problem according to Chapter 8, Problem Management. LMCs will create the MWO in ILM using procedures contained in Chapter 27. The LMC will provide timely feedback to the user/operator on the resolution of the problem. The maintenance role of the LMC includes the following:

(1) Receiving notification of HW problems.

(2) Opening and closing the MWO.

(3) Dispatching the appropriate maintenance technician [system administrator (SA), network administrator (NA), or local or vendor repair technician].

(4) Updating MWO with repair information including problem resolution and identification of failed and replaced parts in a timely manner.

The ILS Maintenance Coordinator is responsible for:

(1) Tracking MWO status.

(2) Reviewing MWO and repair actions for appropriateness and completeness.

(3) Requesting missing MWO information from LMC.

(4) Ensuring the ILM database is updated based on property record actions from the MWO.

(5) Identifying support problem areas.

21.3.1 Hardware Problem Reporting

Once a failure occurs, the operator, SA and/or NA will isolate the problem to its source (i.e., Operating System, COTS application software, ECS custom software, science software, network, or COTS hardware) using the actions in Table 21.3-1, DAAC Hardware Problem Reporting Procedures.

Table 21.3-1. DAAC Hardware Problem Reporting Procedures

Step	Occurrence	Action
1	System problem discovered by an SA, NA, or operator,	a. If local troubleshooting does not fix the problem and it is determined to be hardware related, notify the LMC who will contact the appropriate maintenance vendor. The LMC will start an MWO when the COTS maintenance vendor is notified.

21.3.2 Hardware Corrective Maintenance Actions

Hardware problems are forwarded to the LMC. The LMC will attempt to identify the cause of the problem and employ DAAC resources to resolve the problem. If unable to correct the problem using DAAC resources, the LMC arranges for on-site maintenance by the appropriate maintenance provider in accordance with Section 21.3.4, Contract On-Site Hardware Maintenance.

Hardware corrective maintenance actions are described in Table 21.3-2.

Table 21.3-2. Hardware Corrective Maintenance Actions

Step	Occurrence	Action
1	COTS HW problem not resolved by initial troubleshooting by operator, SA or NA.	a. LMC notified or HW failure by operator, SA or NA. b. LMC opens MWO and adds any cross-reference information for related open Remedy Trouble Ticket. (If existing)
2	LMC attempts to identify cause of problem.	a. Verifies actions and results to date by contacting SA and/or NA. b. Starts corrective maintenance process.
3	Problem resolved by LMC or local staff.	a. If problem can be resolved without hardware replacement (e.g. re-seat component, cable, etc): 1) Correct problem, and verify resolution. 2) Prepare an MWO describing the maintenance action and forward to the EDF. b. If problem can be resolved by replacement of failed LRU with maintenance spare (if available): 1) Replace failed LRU and record following in MWO: a) Part number, serial number, and model/version number of replaced LRU b) Part number, serial number, and model/version number of new LRU c) Down time (elapsed hours/minutes) d) Delay time identified by reason 2) CM requirements are accomplished following procedures in Chapter 9. 3) Order replacement of failed LRU in accordance with Section 21.4.1. 4) Replace the failed LRU in accordance with Section 21.4.2. 5) Prepare a MWO describing the maintenance action and forward to the EDF.
4	Problem not resolved by LMC or local staff.	a. LMC notifies the maintenance contractor by obtaining contractor information from the ILS web page. b. LMC invokes return-to-depot support where appropriate, c. Prepare an MWO describing the maintenance action and forward to the EDF.

21.3.3 Contract On-Site Hardware Maintenance

When on-site hardware maintenance support is necessary, the LMC will notify the applicable maintenance contractor and request assistance. The call for support will be documented in the MWO by the LMC, noting the date and time the contractor was called. It is important that all vendor maintenance activities start and stop times associated with the activities are recorded in the MWO. The MWO is one the primary means of measuring, and managing the maintenance vendor's contractual performance in support of the ECS system availability goals. Data fields have been specifically created in the MWO to capture this information. Refer to Table 21.3.4-1 for more information about obtaining on-site COTS hardware maintenance support.

Table 21.3.4-1. Obtaining On-Site Hardware Maintenance Support (1 of 3)

Step	Occurrence	Action
1	Local support effort did not resolve the problem.	a. LMC gathers information needed to obtain contract maintenance support and contacts the COTS maintenance vendor. Common information that is needed when contacting maintenance vendors is described below. 1) Make, model, serial number, and location of failed systems. 2) Description of problem and symptoms. 3) Criticality of the COTS hardware experiencing the problem.
2	LMC actions	a. Jointly determine between maintenance contractor and site operations staff an acceptable time to bring the equipment down for maintenance [only applicable when entire device is down. Coordination to schedule down time is only required for a functional, but impaired, device]
3	Maintenance technician arrives at the site.	a. LMC arranges for site access using local established procedures. b. If required, LMC requests System Administrator site Help Desk, or other appropriate and authorized personnel to shut down the machine at the predetermined time so that corrective action(s) can begin. Note that any user affected by this action must be notified prior to the machine/system shutting down. c. LMC escorts maintenance technician to the hardware d. Ensures maintenance provider's technician follows all ESD precautions. LMC assists the maintenance technician in resolving the problem. This includes: 1) Arranging for a demonstration of the problem (if possible) 2) Arranging for the equipment to be shut down. 3) Obtaining site available technical references, when needed

Table 21.3.4-1. Obtaining On-Site Hardware Maintenance Support (2 of 3)

Step	Occurrence	Action
3a	Maintenance technician corrects the problem by replacement of parts.	a. If a part is replaced, the LMC accomplishes the following: 1) Obtains from the failed part or the maintenance technician: a) Serial number, equipment identification number (the EIN number on the silver label), and model/version 2) Obtains from the new part: a) Part number, serial number, and manufacturer's model number (if different from part removed, a configuration change request [CCR] is required) 3) Updates the MWO with following information: a) Actions taken to correct the problem. b) Part number, serial number, and model/version, and EIN (if applicable) of the old and new item c) Name of the item replaced d) Arrival date and time e) Time and date corrective action started f) Time and date corrective action completed g) Any delay time experienced in completing the corrective action and reason for delay time to repair
3b	Maintenance technician corrects the problem without replacement of parts	a. If no parts were replaced, the LMC updates the MWO with: 1) Actions taken to correct the problem. 2) Time and date technician arrived 3) Time and date repair was started and completed [these times are required to compute RMA data].
3c	LMC requests the SA to make the system functional	a. SA restores data, operating system, patches or other SW items to bring the system on line. b. Notes the amount of restore time in the MWO.
3d	Maintenance technician does not resolve the problem	a. LMC request the Maintenance vendor provide additional technical and or managerial resource to resolve the problem. b. LMC notifies ILS Maintenance Coordinator that problem repair effort as been delayed, and escalated. c. LMC documents all escalation activity in the MWO until further action is taken. d. Receipt of a completed copy of the dispatch trouble ticket from the vendor e. The information from the vendor's ticket is consistent with the information in the MWO f. Record all actions for future reference and for help in completing the MWO
4	LMC reports actions taken	a. After the failure is repaired complete the MWO and send it to the EDF. b. If appropriate ensures the local and or EMD Configuration Control Board is properly notified of the configuration alterations and requests a formal change using procedures in Chapter 8.

Table 21.3.4-1. Obtaining On-Site Hardware Maintenance Support (3 of 3)

Step	Occurrence	Action
5	LMC files	a. A copy of vendors dispatch sheets, and related documents in a permanent file and references the MWO or files them with copy of the MWO.

21.3.4 Return-to-Depot Support

If the OEM vendor does not provide on site support a return to vendor maintenance concept is used. There are two types of return to vendor maintenance. The ILS web page will specify which concept is to be used. One is advance replacement where the new LRU is requested prior to sending the old LRU to the vendor. If advance replacement is not provided, the LMC must return the failed item to the appropriate repair center using procedures contained in Section 21.4-2.

Table 21.3.5-2 provides the Return to Depot procedures.

Table 21.3.5-2. Procedure for Return to Depot (Advance Replacement and Return before Replacement)

Step	Occurrence	Action
1a	Advance Replacement authorized	a. Contact vendor and request replacement. b. Return broken part to vendor when new part is installed and working.
1b	Advance Replacement not Authorized	a. Contact the maintenance provider using the information obtained from the ILS web page.
2	Ship Failed unit to vendor	a. Obtain an RMA number and shipping instructions from the repair vendor. b. Ship the failed unit to the vendor using local shipping procedures.
3	Replacement Part is received from the COTS maintenance vendor.	a. Place a new EIN sticker on replacement LRU. b. Enters new property information into the MWO c. Install the new LRU. d. Complete the MWO and send it to the EDF.

21.4 Maintenance Spares

The maintenance contractor performing the maintenance will provide replacement LRUs. Major OEM maintenance vendors will have parts depots in the metropolitan area of the DAAC and as such vendor owned site spares are not necessary. However at LP DAAC SGI and STK maintain a minimum number of on site spares at no cost to EMD. These spares may be used for a repair that the LMC is authorized to make. After use the LMC must ensure that the spares are replaced by SGI or STK. If maintenance spares are procured by the EMD ILS Office, based on the requirements of an EMD task order, the spares allocated to the DAACS will be managed at the DAAC by the LMC using guidance from the above referenced documents and appropriate local DAAC policies and procedures.

21.4.1 Use of Maintenance Spares

The EDF maintains a limited number of spares for EMD hardware. A list of EDF spares will be sent to the DAAC LMCs under separate cover. If these items are required email the ILS Maintenance Coordinator and if available a spare will be sent.

21.4.2 Return of Failed LRUs

The LMC is responsible for the return of failed LRUs to maintenance contractors providing replacement parts and not on site support. In such agreements the maintenance provider sends to the site a replacement for a failed component under the condition that the site will return the failed component within a reasonable time, usually not greater than 10 days. If the failed component is not returned the contract is charged the full purchase price for the item not returned.

21.5 Non-Standard Hardware Support

Non-standard COTS hardware support consists of:

a) Maintenance support outside the PPM (Principal Period of Maintenance),

b) Support covered under a Time and Materials contract, or

c) Escalated support actions by the maintenance support provider.

Table 21.5-1 shows the procedure for Time and Material Support

Table 21.5-1. Procedure for Time and Material Support

Step	Occurrence	Action
1	LMC contacts	a. The Time and Material Support Contractor and requests maintenance.
2	Repair is completed	a. After the repair is completed, prepare and MWO and send it to the EDF

21.5.1 Escalation of COTS Hardware Support Problem

In some cases OEM hardware maintenance contractors do not respond to maintenance requirements in a requisite manner that ensures prompt hardware repairs. In these cases the LMC should request escalation from the onsite maintenance technician. If further escalation is necessary the LMC will request the ILS office to escalate the issue within the OEM vendors system. When requesting escalation the original vendor trouble ticket number will be used as the reference document.

21.5.2 Low Cost Equipment – Not Repaired

Server and PC terminals, keyboards, and mice are low cost items that are not repaired, because the repair costs would exceed the cost of a new item, but are replaced on a one-to-one basis by either the OEM maintenance contractor or the time and material contractor. LMCs will follow the same procedures for replacing these items as for other LRUs. If the appropriate maintenance contractor does not remove the failed items, they may be disposed of using local procedures after the MWO is completed and sent to the EDF. Property tags should be removed from the keyboards and mice before disposal.

22. Software Maintenance

22.1 Introduction

The EMD organization provides maintenance for EMD, software, and firmware systems delivered under the EMD contract to the EMD sites.

In general, EMD organizations procure, produce, deliver, and document the modifications, and enhancements made to EMD software and firmware. No custom firmware has been identified as part of the EMD program. Commercial off-the-shelf software (COTS SW), firmware, will be maintained in accordance with the EMD maintenance philosophy for software to provide centralized support for developed items and vendor support for COTS SW.

Specific software support procedures are discussed in this section. EMD Project software consists of COTS, custom-developed, and science software. Science software, developed for use on the EMD project, is the responsibility of the science community.

COTS software maintenance includes:

- License to use COTS support contract with the software vendor for telephone assistance in resolving COTS SW problems, as well as obtaining patches and upgrades.
- Services required to produce, deliver, integrate, install, validate and document modifications of existing EMD software and firmware. [The DAAC maintenance activity includes: software configuration management (CM) including support for change control, configuration status accounting, audit activities, and software quality assurance (QA). Each site is the CM authority over its own resources subject to EOSDIS delegation of roles for EMD management.]

The Integrated Logistics Support (ILS) COTS Software License Manager will notify DAAC LMC, System Administrator (SA) and Network Administrator (NA) on the procedures for handling COTS software upgrades and vendor liaison.

The EMD COTS SW Team provides assistance when COTS software issues exceed the capabilities of the site System Administrator or the Network Administrator to resolve.

22.1.1 COTS Software Maintenance

Operations personnel at the sites accomplish installation of patches, upgrades and software problem isolation. The COTS software vendors support COTS software procured for the EMD contract. (The term software vendor refers to the company having the legal right to authorize software use and to modify the software code.) COTS software vendor support consists of telephone support for resolution of usage and interface problems, access to an on-line solution database, providing upgrades and patches and resolving COTS software code problems.

The Activity Outline in Table 22.1-1 is an overview of COTS Software Support procedures and the section number where details for performing the tasks can be found.

Table 22.1-1. COTS Maintenance - Activity Outline

Task	Section
Assist System Administrator in obtaining COTS SW support	22
Manage COTS Software Maintenance Contracts	22.1.2
Manage COTS Software Licenses	22.1.3
COTS SW installation and upgrades	22.1.4
Obtain COTS Software Support	22.1.5
COTS SW Problem Reporting	22.1.6

22.1.2 Management of COTS Software Maintenance Contracts

The EMD procurement office at the EDF contracts COTS software vendor support. After the first year of warranty support, support is contracted for a period of one or more years and extended or modified as operationally required. Information related to COTS software support contracts is maintained in a database used by the ILS COTS Software License Manager to monitor the expiration dates and contract terms.

EMD COTS SW maintenance is purchased annually during the fall of the year. The COTS SW license administrator works in conjunction with the COTS EMD procurement function and COTS SW engineers to determine the products, level of maintenance and quantity of licenses to be renewed. All COTS SW maintenance recommendations are approved by the SITE and EMD Chief Architect. The ILS COTS SW License Manager is coordinator to ensure that all products are identified and worked during the maintenance renewal period.

22.1.3 Management of COTS Software Licenses

Functions of the ILS COTS SW License Manager include the following:

a. Maintain accountability for all COTS SW licenses procured for the EMD contract. Accountability includes tracking and reporting the as-installed location of all licenses (other than PC based products) procured for the EMD program.

b. Assist the COTS SW organization in impact analysis of proposed COTS SW upgrades and patches on other COTS SW applications incorporated in the EMD system design. Maintain a database containing license keys of project-purchased COTS SW. The COTS SW installation team (or site) will provide the host identifications to the ILS COTS SW License Manger, who will then obtain the necessary license keys from vendors for SW installation and populating the COTS SW database.

c. The COTS SW organization will distribute SW upgrades, with vendor-provided release notes.

d. Keep COTS SW and all other EMD sites informed by providing them with the vendor maintenance necessary to access vendor patch libraries for use in resolution of software problems. COTS software licenses vary by the type of software and the software vendors' policies.

COTS software license types include: floating, per site, specific number of concurrent users, unlimited users, and lifetime use without regard to number of users or location. The EMD procurement office purchases the type and quantity of software as determined by the EMD design engineers. COTS software licenses are received and entered into the License Management Module of ILM by ILS COTS SW Manager. The ILS Property Administrator maintains the master copy of COTS SW license agreements (hardcopy).

22.1.4 COTS Software Installation and Upgrades

The COTS software upgrades are subject to appropriate CCB approval before they may be loaded on any platform. The COTS Software Librarian, using procedures contained in Chapter 9, "Configuration Management," notifies the COTS SW organization of the upgrades that have been received. The COTS SW distributes the COTS software upgrades as directed by the CCB. The site System Administrators are responsible for upgrading the software on the host machine and providing follow-up information to the Configuration Management Administrator (CMA), COTS Software Librarian and the COTS SW License Administrator.

COTS software patches may be provided by the COTS software vendor in response to a DAAC's call requesting assistance in resolving a COTS software problem. The problem may or may not exist at other locations. When a COTS software patch is received directly from a COTS software vendor (this includes downloading the patch from an on-line source), the DAAC's CCB shall be informed via CCR prepared by the appropriate site personnel. It is the responsibility of the appropriate site personnel to notify the CCB of the patch's receipt, purpose, and installation status, using procedures contained in Chapter 9, "Configuration Management," and to comply with the CCB decisions. The appropriate site personnel will install the COTS SW patches as directed by the CCB. In addition to providing patches to resolve problems at a particular site, the software vendor will periodically provide upgrades of COTS software in order to improve the product. These upgrades are issued to all licenses covered by a software maintenance contract. Therefore, the COTS software upgrades will be shipped to the ILS Property Administration (PA) who receives and enters them into inventory and then forwards the media to the COTS Software Librarian and the licenses/keys to the COTS Software License Administrator. The COTS SW team will direct COTS SW deployment activities.

22.1.5 Obtaining COTS Software Support

COTS SW support involves both site capability and contracted support. The site System Administrator (SA), Network Administrator (NA), and site Software Maintenance Engineer provide site capability. The COTS SW vendor provides contracted support. When the appropriate site personnel confirm that a problem is attributed to the COTS SW, the COTS SW vendor's technical support center is contacted by authorized personnel at the site.

The software vendor's technical support center will verify contract support authorization and then assist in pinpointing the COTS SW problem to provide a recommended solution. The solution may involve a patch or work-around, or the fix may be included in a future release. If a patch exists to correct the problem, the patch will be identified and provided by the software vendor over the Internet or mailed to the requester. If a patch is required but unavailable, the site and vendor together determine the seriousness of the problem. In cases where the problem is critical, a temporary patch or work-around may be provided. If non-critical, the solution to the software problem may be scheduled by the software vendor to be incorporated in a future update or release. (NB: The DAAC and ESDIS CCBs must authorize the patch to be installed as a permanent installation. This decision may be made after-the-fact. That is, if the patch is needed in order to proceed with operations, notify the appropriate DAAC personnel of the requirement in accordance with Chapter 9, "Configuration Management." Applicable requirements of Section 8.3, "Using the Trouble Ticket System," must be followed.) The ILS COTS Software License Manager obtains the support authorization codes from the vendors and arranges for specified personnel to become an authorized contact person, based upon the limitations imposed by the vendor, and the needs of individual DAACs. The software vendor's technical support telephone numbers, the names of personnel authorized to contact the vendor, and the authorization codes will be provided to the site's LMC by the ILS COTS Software License Manager through the ILS web page:

http://dmserver.gsfc.nasa.gov/ils/intro.html

22.1.6 COTS Software Problem Reporting

The first person experiencing or observing a potential COTS SW problem will initiate a trouble ticket according to the procedures found in Section 8.3, "Using the Trouble Ticket System" then forward it to the assigned site person to review the problem. This person will attempt to isolate the source of the problem to system configuration, hardware, network, COTS SW, custom SW, or science SW.

If it is confirmed to be a COTS SW problem, the authorized contact person should contact the vendor's technical support center for assistance. Information on contacting the software vendor's technical support center is in Section 22.1.5, "Obtaining COTS Software Support." The appropriate site personnel must annotate all actions inclusive of dates, time, resolutions, and comments in the Remedy Trouble Ticket as the repair progresses. COTS software corrective action reporting follows the procedures contained in Chapter 8, "Problem Management" and the configuration control requirements contained in Chapter 9, "Configuration Management," when a configuration item is removed and/or replaced with a different version or release.

One method of troubleshooting the COTS SW problem is to scan the software vendor's web site solutions database to learn of any solutions for similar problems. The software vendor's web site address can be obtained as stated in Section 22.1.5. Another manner of troubleshooting the COTS SW problem is to exercise any software diagnostic routine embedded or downloadable that will determine the status of the COTS SW on the equipment by reviewing the troubleshooting-diagnostics and corrective actions taken to date. These troubleshooting, diagnostics, and/or isolation procedures may be contained in the vendor's operational manuals or in locally devised troubleshooting procedures.

COTS SW problems that cannot be corrected using site and contracted software support may be escalated to the EMD SSO. The SSO is staffed with Senior Systems Engineers knowledgeable on COTS SW that can assist in diagnosing the problem.

The site Local Maintenance Coordinator may go directly to the software vendor or to the ILS SW Administrator to obtain an escalation of software vendor support if the software vendor's efforts have not produced satisfactory results within a reasonable period of time. The escalation may result in increased vendor management review of the problem resolution, the assignment of additional resources to resolve the problem, and/or a more highly qualified technician assigned to resolve the software problem.

This page intentionally left blank.

23. Property Management

This section describes procedures for the receipt, control, and accountability of EMD property at EMD sites. The Property Management Plan (PMP), 105-EMD-001, is the DCMA Property Administrator approved document that addresses the process and policies regarding how EMD property is to be managed. The site Local Maintenance Coordinator (LMC) at each site is the site's property administrator. LMCs should be thoroughly familiar with and comply with the policies and procedures contained in the Property Management Plan.

The LMCs support the activities of receiving, inspection, storage, issue, inventory recording, accounting, and reporting of EMD property at EMD sites. As stated above LMC should follow the particular procedures of the PMP; local procedures may used as long as they do not deviate from the specific direction in the PMP. Locally developed procedures should be forwarded to and reviewed by the EMD Integrated Logistics Support (ILS) Office for consistency with this plan.

23.1 Receipt of Equipment and Software from Vendor

Most EMD property will be shipped to DAACs from the EDF at Landover. However there may be occasions when property will be shipped directly to the DAACs from OEM vendors. In such cases, the ILS Property Administration (PA) will fax a copy of the Purchase Order to the LMC to serve as a due-in notice. Upon receipt of the equipment, the LMC will perform a receiving inspection to verify correctness of delivery, quantity received and to determine if items were damaged during shipment. The LMC will utilize the following tables as guidance for Receipt of Incoming Items.

- Table 23.1-1, Procedure for the Receipt of Equipment
- Table 23.1-2, Procedure for Completion of the Inventory Worksheet
- Table 23.1-3, Procedure for Completion of the Non Conforming Product Report
- Table 23.1-4, Receiving Process Checklist
- Table 23.2-1, Property Received from the EDF

The worksheet for documenting inventory as well as the checklists can be located on the Web at http://dmserver.gsfc.nasa.gov/forms/formindex.html under Project forms. Inventory Worksheet form number is Mo05ja99.doc. The Property Checklists form number Mo06ja99.doc, which includes the Loading Dock Checklist, System Verification Checklist and Receiving Process Checklist. The ILS Property Administration has incorporated these two forms together as one document to reduce paperwork. When all checklists are signed and verified, the LMC will fax or email all forms to the ILS Property Administration and ILS Property Administration will update the Inventory Database. When a product is received that does not conform to the purchase order the Non-Conforming Product Report (NCR) form located at the same URL with the following form number Mo08ja99.doc can be accessed for use.

Table 23.1-1. Procedure for the Receipt of Property

Step	Action
1	LMC completes the Loading Dock checklist document with the following information: Printed name of receiving individual Signature of receiving individual Date of receipt Name of the carrier Shipper's bill of lading or tracking number Customer reference number (when appropriate) Number of boxes received Condition of boxes with a notation of Satisfactory or Damaged
2	LMC verifies damage, shortage, overage or other discrepancies and annotates these findings on both the carrier's and site's copy. A NCR will be competed per the instructions in Table 23.1-4.
2a	If there is obvious damage to exterior packaging at the time of delivery, obtain a signature of the carrier's representative on the shipping carrier's document and notify the ILS PA for further direction and information.
3	Begin the process of moving equipment into a controlled storage area and completing the inventory worksheet as described Table 23.1-2.

Table 23.1-2. Procedure for Completion of the Inventory Worksheet

Step	Action
1	LMC removes the equipment from the loading dock to controlled storage area.
2	Verifies the items received against purchase orders and vendor's packing list. Inspects visual condition of material and documents information on Inventory Worksheet.
3	Documents the serial, model numbers and other appropriate markings on the Inventory Worksheet. Performs final visual inspection of product to ensure no damage or non-conforming items have been received.
4	Places silver EIN tags on equipment per the instructions listed in Section 23.2.
5	Assembles equipment for burn-in and documents the burn-in process on the System Verification Checklist.
6	If required, complete a NCR per Table 23.1-3.

Table 23.1-3. Procedure for Completion of the Non Conforming Product Report

Step	Action
1	LMC verifies shipment discrepancies (include shortages, overages, and incorrect items/quantities/models.
2	Completes COTS NCR per instructions on the back of the form.
3	Sends the completed COTS NCR to the ILS PA via fax or EMAIL.
4	The ILS PA will process the NCR at the EDF and notify the DAAC of resolution.

Table 23.1-4. Receiving Process Checklist

Step	Action
1.	Verify that Loading Dock Checklist has been completed with all the appropriate information.
2.	Annotate Purchase Order number or Returned Material Authorization (RMA) on the Inventory Worksheet.
3.	If required, verify that COTS NCR has been completed and processed per Table 23.1-3.
4.	If the material is partial receipt, verify that it is segregated, labeled, marked and in a controlled storage area.
5.	If the order has discrepancies and an NCR has been forwarded to the EDF, verify that the property is segregated, labeled and documented in a controlled storage area.
6.	Verify the Inventory Worksheet, System Verification, and Receiving Process Checklist have been completed and documented. Fax or Email to the ILS PA.

23.2 Receipt of Equipment and Software from the ILS Property Administrator

The LMC will also receive equipment from the ILS PA. Table 23.2-1 defines the process of receiving EDF shipped equipment at the DAACs.

Table 23.2-1. LMC Actions for Property Received from the ILS Property Administrator

Step	Action
1	LMC receives shipment with Installation Receipt (IR) report from the ILS PA.
2	LMC inspects and verifies for shipping damage, completeness using the IR report.
3	Notifies the ILS PA immediately of any discrepancies. If discrepancies exist, annotate the IR report accordingly.
4	Sign for the property where indicated and redline location and site-specific changes on the top right header information.
5	LMC retains a copy for files and mails the original back to ILS PA.
6	The ILS PA enters the redline changes into the ILM Property Database.
7	The original signed redlined IR report and a copy of the Installation Report will be filed in the EMD equipment folder.
8	The site copies of the IR report and installation report will be filed in the site equipment folders.
9	The LMC will coordinate installation of the equipment in accordance with approved CCR and DAAC procedures.

23.3 Equipment Tagging

EMD equipment (e.g., contractor-acquired and GFP) that is separately identifiable and meets the criteria for controlled equipment as described in the Property Management Plan, will be tagged with EMD property tags. Figure 23.2-1 illustrates EMD property tags. LMCs may use either EMD or ECS property tags. ECS property tags shown below. EMD property tags are the same except the contract number is NAS5-030398. All property shipped from the EDF will be tagged prior to shipment. If LMCs have to tag or retag EMD property tags will be placed on the equipment so that they are visible and easily accessible by bar-code scanners. Vendor-loaned and RSC capital equipment will not be tagged with EMD property tags.

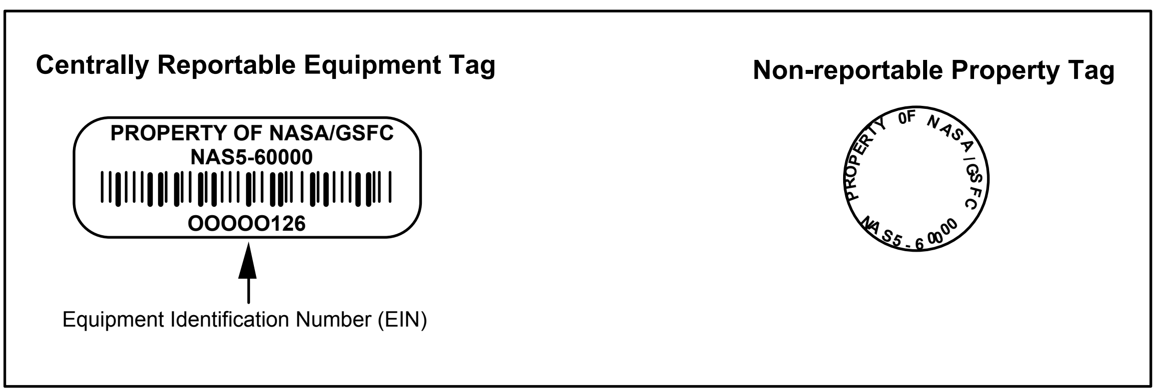

Figure 23.2-1. EMD Property Tags (Actual Size)

Components of major equipment that are not separately identifiable or stocked for use, such as spares/repair parts will not be assigned an EIN. Items not given an EIN sticker will be controlled as inventory items and recorded by manufacturer, description, model/version, serial number (if applicable), location, and quantity and with the parent EIN.

Property tags of loaned GFP equipment containing a NASA equipment control number (ECN) will not be removed by the LMC. At the time of receipt of such property, the ILS PA will affix an EMD property tag with EIN next to the government tag.

If EMD property is returned to a vendor for repair or replacement the LMC will remove the EMD property tag prior to shipment. When the item is returned the LMC will place a new property tag on the property and report the new tag, serial number and other indicative data to the ILS PA. The LMC may use either an MWO or an email message to report the new data.

23.4 Property Records and Reporting

The EMD Project will use the Inventory Logistics and Maintenance (ILM) system to support the property management, control and reporting functions for contractor-acquired and GFP equipment. Property records are created and maintained in this system by the ILS PA. LMCs will be able to do queries and prepare reports using the system. Instructions for ILM use are in Chapter 27 of this document.

Property records will contain a line for each item having an EIN (e.g., workstation monitor) and each of its major components (e.g., network interface cards, RAM chips, graphics card). Refer to the IR Report provided at the time of site installation. **It is the responsibility of the LMC to notify the ILS PA of inventory, configuration, and location changes so that site property records will be maintained current.** See Section 23.5 for specific instructions.

23.4.1 Maintaining Property Records

LMCs will document inventory and configuration changes in local property records within one business day of the change and appropriately notify the ILS PA.

Support documentation for posting changes to property records include the following:

- **Installation Receipt Report** -- This report is provided by ILS PA or the installation team at the time equipment is installed and is used to record receipts and changes of equipment at the site. It can be used to update site property records with installed location, date, and name of the person accepting receipts. This report is signed by the LMC to acknowledge receipt of equipment at the site.
- **Maintenance Work Orders (MWO)** -- Prepared by the site LMC to report equipment changes resulting from maintenance or relocation actions (e.g., serial/model changes, component replacements, and relocation/reconfiguration at the site). MWOs are used by the ILS staff to update the ILM property records.

23.4.2 Reporting Loss, Theft, Damage or Destruction

If EMD or GFP property at the site is lost or stolen, the LMC will notify the site security manager as soon as the theft is discovered and the ILS PA within one business day. The initial written report will contain all information related to what was lost/stolen, when, where, how, and the circumstances regarding the loss/theft. The final report, due 30 days later, will contain all information required by the EMD Property Management Plan and will be signed by the DAAC Operations Manager. If a report was prepared by local security personnel/police, a copy of the report should be attached to the report.

If equipment is damaged or destroyed through circumstances that indicate inappropriate use, negligence, or improper care, the LMC will likewise notify the ILS PA of the specifics of the damage/destruction and its circumstances via Internet as soon as known.

The ILS PA will review such incidents and report, as required, to the EMD Contract Manager. The ILS PA and the EMD Contract Manager will notify the DCMA Property Administrator of any theft or destruction of EMD property.

23.5 Equipment Relocation

This section provides instructions for equipment relocation within a DAAC (intra-site relocation); between EMD sites, and between EMD sites and non-EMD sites (inter-site relocation); to a vendor (off-site relocation); and transfer to outside the contract (external transfer).

23.5.1 Intra-Site Relocation

Requirements for equipment reallocations within the facility or between facilities at the same site will be processed through the LMC to maintain control and accountability of equipment inventories. Local procedures will be developed to ensure that the LMC is aware of all property moves within a DAAC. When completed, the LMC will report the location change via a MWO, EMD Move Change Form Email to the EMD PA.

23.5.2 Inter-Site Relocation

Inter-site relocation requests require a CCR approved by the EMD CCB. Such requests will identify by EIN and equipment description what is to be moved, where and when it is to be moved, and the reason for the relocation. The EMD PA will work with the loosing and gaining DAAC to ensure that all property information affected by the transfer is update in ILM. The losing site LMC will coordinate the relocation resources and schedule with the gaining site and the ILS PA. Once completed, the gaining site’s LMC will report completion of the relocation to the ILS PA by a MWO or Email. Any loss or damage to the equipment will be reported using the procedure described in Section 23.3.3 when it occurs or is first discovered.

23.5.3 External Transfers

Generally EMD property will not be transferred outside the EMD contract. If LMCs feel they need to transfer EMD property to organizations not support by EMD, contact the EMD PA for guidance. No property will be transferred without the express approval of the EMD PA.

23.6 Inventories and Audits

Annually LMCs will complete a 100 percent physical inventory of controlled EMD property and GFP at the site. This inventory may be conducted in conjunction with the Task 101 annual inventory conducted by the ILS office. EMD personnel responsible for maintaining property records will provide technical assistance but will not be part of any inventory count teams. Inventories will be designed to achieve the following objectives:

- Verify that accountable equipment is still on hand.
- Confirm or determine current locations and custodial responsibility for equipment and material.
- Identify unrecorded equipment that qualifies for control.
- Locate or identify missing equipment.
- Identify unused or under utilized equipment and equipment or material in need of repair or rehabilitation.

If the annual inventory is done independently of the annual ILS inventory the LMC will, at the time of completion of the site directed inventory, forward a copy of the Inventory Reconciliation Report to the ILS PA. The Inventory Reconciliation Report will be signed by the site's EMD Manager attesting that a 100 percent inventory was conducted and that all equipment is accounted for except for those indicated as not on hand. All discrepancies will be explained.

23.7 Storage

Access to equipment and software in storage will be limited to authorized personnel and controlled by the LMC. LMCs will ensure that storage areas are kept in a clean, orderly manner. Material will be stored on shelves, in bins or drawers as appropriate, and its storage location entered into the site property record. Special storage areas or controls will be provided for items subject to corrosion, humidity, and temperature. LMC should ensure that Electrostatic Discharge (ESD) procedures are used for all items requiring ESD protection. See Section 23.9 for specific ESD instructions. Such items will be inspected semi-annually by the LMC. Serviceable property does not require any special color tag.

23.7.1 Segregation Requirements

Contractor-owned and vendor-loaned property will be segregated from EMD Government-owned property during storage. Unserviceable equipment will also be segregated from serviceable equipment and will be tagged. Unserviceable/reparable equipment will have a yellow tag affixed to it; unserviceable/non-reparable equipment will have a red tag affixed. Unserviceable equipment tags will indicate reason item is unserviceable, date it became unserviceable, parent EIN it came from, and signature of person declaring the item unserviceable.

23.7.2 Stock Rotation

EMD is a task order contract and Task 101 does not provide for the EMD to provide DAAC distribution and other consumables, however DAAC purchase consumables such as computer tapes, cleaning tapes, CDs, DVD, labels, etc., should be used on a first-in, first-out basis.

23.7.3 Physical Security

EMD property will be stored in secured areas where access will be limited to authorized personnel and controlled by the LMC.

23.8 Packing and Shipping

Prior to shipping centrally reportable equipment to the EDF or other EMD sites, the shipping LMC will notify the gaining LMC of the shipment by email. The email should include the expected shipment date, carrier, shipping document number, estimated weight and cube, number of pieces, shipper and ship-to-address. Prior to shipment, a pre-shipment inspection will be performed to verify the following:

- Correct identification of equipment on packing lists and shipping documents including configurations, serial numbers, number of containers, and ship-to address.
- Adherence to packing, packaging and marking standards.
- Inclusion of appropriately prepared documents within shipping containers.
- The gaining LMC will notify the EMD PA via Email or phone when the item shipped has been received.

EMD property being shipped from vendors and the EDF will be shipped to the DAAC facility to the attention of the DAAC Local Maintenance Coordinator. Local policy at some sites may require delivery to a site central receiving point. In such cases, written procedures will be developed between the LMC and the site's central receiving office regarding notification of receipts, documentation required, and provisions for local delivery to the DAAC facility. The delivery of EMD equipment to site central receiving points versus direct delivery to the DAAC facility will be determined based on agreements and procedures established between the host facility and the DAAC.

23.9 Electrostatic Discharge (ESD) Program

An Awareness program on Electrostatic Discharge (ESD) and operations and maintenance practices will be followed to eliminate ESD hazards to HW, SW, or people. Procedures for the program will be developed using DOD-HDK-263 and DOD-STD-1686 as guides. Included in the program will be policies and procedures for prevention and safe dissipation of static electricity: Workplace common grounding requirements; and parts handling and protection when in storage, outside the manufacturer's protective packaging, and being readied for installation or removal and packaging for delivery. ESD hazard awareness and prevention will be an appropriate part of the training and certification process of EMD operations and maintenance personnel. All ESD hazard awareness and prevention requirements will be passed through as requirements to all operations or maintenance subcontractors.

24. Installation Planning

Installation and Facility Planning is an integral part of the EMD Task Order Delivery Process. When an EMD Task Order is received from NASA the installation and facility planning cycle begins. The hardware installation planner in conjunction with hardware engineering will review current floor plan, hardware diagram and electrical plan. These plans identify space, power, and cooling requirements based on design information available at the time the Task Order is executed. The Facilities Plan identifies facility preparation requirements and general installation planning that is based on final design information. The Installation Plans provide the detailed planning required by installation teams and the sites to make final preparation for installing Release equipment. Close coordination with the DAAC is required to ensure that the projected plans meet DAAC requirements. While site surveys are not required, one may be performed based on the complexity of the task described by the Task Order.

24.1 Responsibilities

Installation planning and coordination is the responsibility of the ILS Installation Coordinator, who is part of the Integrated Logistics Support (ILS) Office. Using information from telephone calls or site surveys, the Installation Coordinator prepares the Facility Plans and the Installation Plans and coordinates actions needed to prepare for and conduct the installations. DAAC personnel support the Installation Coordinator by working to prepare DAAC specific plan for each Task Order. They also ensure that site preparations/coordination are completed on schedule and work with EDF installation technicians to complete the installation of hardware.

24.2 Process Description

DAAC site surveys have been previously conducted to obtain DAAC-specific information needed to begin the installation planning process. The DAAC information, plus design and equipment specifications, is used to prepare the Facilities Plans, which project facility requirements and provide a preliminary plan for the placement of systems at the DAACs. DAAC engineering staffs review this information and provide requested changes, which are incorporated into the Installation Plans.

Prior to the installation of hardware, a detailed Installation Plan is produced to identify the planned placement of hardware in the facility and how the hardware will be configured and networked, and to identify site preparations necessary to support the installation. Installation teams use the Installation Plan to install the systems and networks. Within three weeks following the installation, the Installation Coordinator will update the floor plans to reflect the as-installed configuration at the site. These diagrams are submitted to the ECS CCB and, when approved, becomes part of the operations baseline for the site and are available for viewing on the web. It is the responsibility of the LMC to notify the Installation Coordinator as changes to the baseline documentation occur. Table 24.1-1 shows the Installation Planning Activity Outline.

Table 24.1-1. Installation Planning Activity Outline

Step	Responsible Person	ACTION
1	Installation Coordinator	In conjunction with the DAAC staff prepare an installation plan.
2	Installation Coordinator	Briefing with DAAC SE and coordinate schedule
3	Installation Team	Install hardware based on Installation schedule
4	Installation Team	Burn in and test new hardware
5	Installation Coordinator	Update information to the plan and create a As-Built document consisting of Floor Plan and Hardware Diagram
6	Installation Coordinator	Send the As-Built document to DAAC Liaison while creating a CCR for the CCB
7	Installation Coordinator	Prepare CCR and board the new floor plan and hardware diagram.

24.3 Maintenance of Facility and Hardware Diagrams

Facility and hardware diagrams reflect the as-installed configuration. The baseline version of the floor plan is maintained by the ILS Installation Coordinator, and the hardware diagrams are maintained by the hardware engineering staff. As changes to these diagrams occur (e.g., relocation of equipment within the site), the LMC will inform the ILS Installation Coordinator or responsible hardware engineer of changes. The Installation Coordinator or responsible hardware engineer will update the appropriate documents and create a CCR to present the changed documents to the EMD CCB to update the documents.

25. COTS Training

This chapter outlines the procedures for DAACs to request COTS HW and SW training from the EDF. The EMD program has limited training funds to provide training to DAAC and EDF personnel. Operations Contractors at DAACs are generally not eligible to use these training dollars. These funds are reserved for Raytheon and Raytheon subcontractors. The COTS training funds are not generally used to fund travel; exceptions will be made on a case-by-case basis. DAAC Liaison Leads will contact the Integrated Logistics Support (ILS) training coordinator directly to request training.

The Activity Checklist in Table 25.1-1 outlines the role of the COTS Training Coordinator and the section number where details for performing the tasks can be found.

Table 25.1-1. COTS Training - Activity Checklist

Task	Section
Requesting COTS Training	25.1
Coordinating COTS Training	25.2
Canceling/Rescheduling COTS Training	25.3
Contractor COTS Training Funds Accounting	25.4

25.1 Requesting COTS Training

DAAC Leads will submit training requests to the Senior DAAC lead and the ILS Training Coordinator at the same time. It is understood that unless the Senior DAAC lead specifically disapproves the training it is considered approved. The ILS Training Coordinator will wait two business days and then process the training request. The request for training should be sent at least 30 days prior the requested training date.

The following steps outline the process:

a. The training request will contain the following information:
   1. Student name(s) and DAAC representation.
   2. Training need.
   3. COTS course requested.
   4. Dates preferred.
   5. Price of COTS course.
   6. Manager approving purchase of training.
   7. Course location.
   8. Duration of course.

b. The ILS Training Coordinator verifies that the training request meets the following criteria:
   1. Meets the appropriate company/location criteria.
   2. Relates to COTS hardware or software in the EMD system design.
   3. Is cost effective and within budget constraints.

c. COTS Training Coordinator determines the proposed training details, including the following:
   1. Training vendor.
   2. Individual or group training, based on cost effectiveness.
   3. On-site or off-site class location.
   4. Available vendor training dates.

d. The ILS Manager (prior to procurement) receives the training request (forwarded by the COTS Training Coordinator) for approval. The ILS Manager will either approve or deny the request.

e. COTS Training Coordinator maintains record of approval of training purchase.

25.2 Coordinating COTS Training

After the ILS Manager approves the request for COTS training, the COTS Training Coordinator will work with the COTS procurement office to schedule and procure the training. The procedures to purchase training are accomplished in the following order:

a. When approved, the COTS Training Coordinator submits all training details to the COTS Purchasing Manager.

b. The COTS Training Coordinator orders training from the vendor.

c. Purchasing Manager processes the purchase order and provides a copy to the COTS Training Coordinator.

d. The COTS Training Coordinator forwards the purchase order to the vendor to reserve training.

e. The COTS Training Coordinator generates a notice to students that includes training vendor, course, date(s), and other relevant information.

f. For on-site training, at the EDF, the COTS Training Coordinator makes necessary arrangements for classroom space and equipment configuration; coordinates use of any operational equipment required for course, with on-going operations; forwards site location details to vendor instructor.

g. Prior to Group COTS training, the COTS Training Coordinator provides students with a COTS Training Evaluation Form, which evaluates the effectiveness of the course. In cases when COTS training is found to be substandard or ineffective, the COTS Training Coordinator contacts the ILS Manager, and the DAAC Senior Lead so they come to a consensus as to whether or not to pursue compensation for the training.

h. Depending upon the decision rendered, the COTS Training Coordinator seeks refund, replacement training seat, or training credit from the vendor.

25.3 Canceling/Rescheduling COTS Training

COTS training vendors generally withhold all or part of registration fees for course seats canceled too close to the start date of training. The deadline for cancellation without penalty varies between vendors, **but the maximum deadline is three weeks prior to course start date**. In order to preserve EMD COTS training funds, any cancellations of COTS training by EMD personnel must be made before three weeks of the start date to avoid these financial penalties.

- If student(s) need to cancel within this three-week deadline, the DAAC Lead will be responsible for substituting an equally qualified individual to attend the course, and for notifying the COTS Training Coordinator to ensure proper record keeping and registration with the vendor.

25.4 Contractor COTS Training Funds Accounting

COTS training funds will be allocated to each EMD DAAC and EDF organization, based upon staffing levels and functions performed at the site. The ILS COTS Training Coordinator maintains the training budget spreadsheets for each of the DAACs and EDF. The ILS COTS Training Coordinator updates the spreadsheets as training is complete and submits monthly reports the ILS manager and DAAC Leads.

While the coordination and purchasing responsibilities for COTS training fall primarily with the ILS Manager, the DAAC Senior Lead will provide advice and consent in to ensure the training dollars are allocated and spent in the most efficient and effective manner for the EMD program.

This page intentionally left blank.

26. Science Software Integration and Test (SSI&T)

The purpose of this section is to delineate the operational procedures to accomplish the various steps that may be involved in the integration and test of Science Data Production Software (SDPS/W) with the Earth Observing System (EOS) Data and Information System (EOSDIS) Core System (ECS). The SDPS/W integration and test (SSI&T) is performed at the Distributed Active Archive Center (DAAC) responsible for the generation of the standard products. The SSI&T Operational Procedures for the ECS Project, 162-TD-001, (aka The Green Book) was the main reference for this section. General information concerning preparing and delivering SDPS/W to the DAAC is found in the Science User's Guide and Operations Procedure Handbook for the ECS Project, Part 4: Software Developer's Guide to Preparation, Delivery, Integration and Test with ECS (205-CD-002). Each DAAC and Instrument Team (IT) combination has formulated specific agreements, understandings, or procedures that will guide their respective SSI&T activities.

The procedures in this document provide detailed instructions on how to use the tools that are provided in ECS to accomplish the steps outlined in the DAAC-IT procedures. The SSI&T operational procedures are given in this section. They are organized by activity. The order in which the procedures appear loosely follows the order in which they will usually be performed. It is highly recommended that a copy of the ECS Internal Interface Control Document for the ECS Project (313-EMD-001) be readily available. This document contains more specific details for SSI&T that apply to the many variants of ECS processes that you may encounter.

These procedures present the use of GUIs. Some procedures may have a command line equivalent; these are documented in the corresponding GUI help screens but are not presented here in the interest of simplicity. The Operations Tools Manual for the ECS Project, 609-EMD-001 should be referred to for more detailed information on how to use GUIs and command line equivalent usage.

A Note about the Order of Procedures

The science software I&T operational procedures contained within this document are ordered in a manner that is intended to *loosely* suggest a logical sequence which, when used as a "road map", represents an overall, sensible end-to-end SSI&T activity at the DAACs. The ordering cannot, however, be interpreted as a detailed, step-by-step guide to SSI&T activities. In addition, since there are many factors that affect the actual SSI&T activities (e.g. Instrument Team deliveries, DAAC policies, agreements between the Instrument Teams and the DAACs, etc.), the ordering in this document can only be suggestive.

Many of the procedures outlined in this document are inter-related. A procedure may assume that another procedure has been completed. In general, the ordering of the procedures reflects this. The user should be aware, however, that this is not the case for all procedures. Therefore, depending on the SSI&T activity, the ordering suggested may not apply. Procedures may require other procedures that appear *after* the procedure requiring them.

Assumptions:

1. All procedures in this section assume the following: that the Instrument Team has delivered the science software to the DAAC and that the most recent release of ECS is available at the DAAC.

The following conventions are followed for explaining procedures:

Text that should be typed literally in the "Action" column of the procedures is displayed in `courier` font. Text within a literal command that represents a fill-in-the-blank object is displayed in ***`italic courier`*** font. (Example: **`cd`** ***`mydir`*** means type "cd" and then type the name of the correct directory.)

A command line in the "Action" column that should be typed in without a line break will be indicated by an indent in any following lines. The end of the command is indicated by **<ENTER>,** which stands for pressing the **ENTER or RETURN** key.

26.1 DAAC Science Software I&T Support Engineer

Provide DAAC SSI&T execution support, ECS tool and system expertise and science S/W processing problem support. Provide support to scientists in the development and integration of science software for both updates and new science software into the DAAC ECS system.

26.1.1 Interfaces

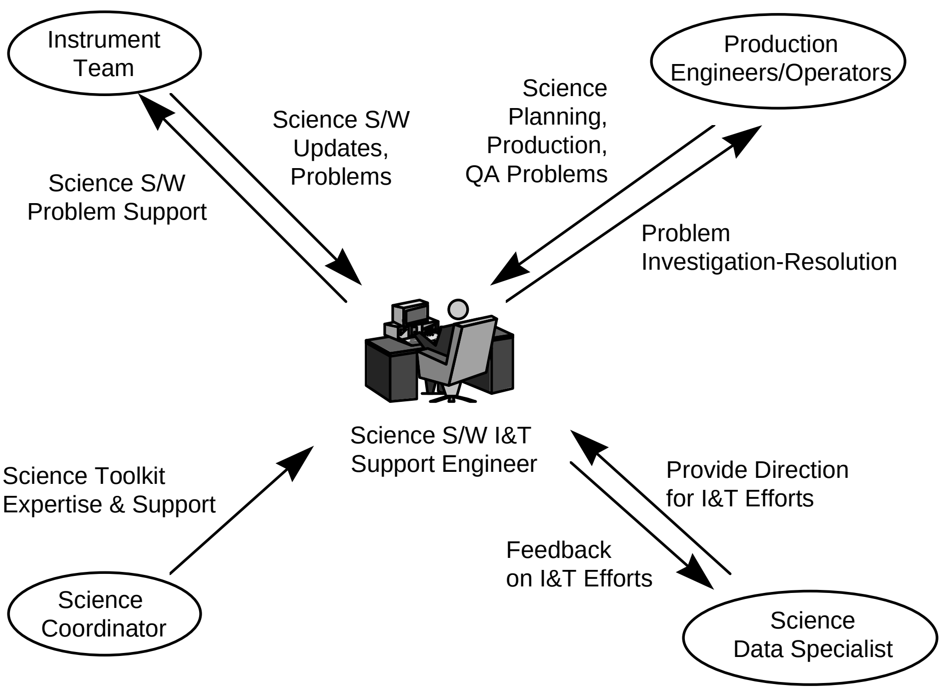

Figure 26.1-1. DAAC Science Software I&T Support Engineer Interfaces

26.1.2 Roles and Responsibilities

1. Provide support to Instrument Teams for the development and integration of science software into the DAAC ECS system.
2. Perform standards checking on all delivered software, including source code, scripts, process control files and related documentation.
3. Provide support for metadata updates and additions for science data products.
4. Support science processing problem investigation and resolution.
5. Recommend - assess, develop, implement changes to science toolkit software.
6. Provide support for integration and test of new and modified toolkit functions into the science software.
7. Support and track configuration of delta changes to science software, calibration coefficient files, relevant input files and documentation.
8. Facilitate migration and integration of major and minor modified versions of the science software into production versions.
9. Assess impacts and support integration and test for production planning and processing software changes. (Planning Workbench GUI)
10. Provide feedback and receive direction on I&T efforts.
11. Recommend-assess improvements for automated tools for SSI&T activities, such as file comparison and viewing data.
12. Write and implement procedures to examine non-standard auxiliary files and files not in HDF EOS format.

26.1.3 Science Software Integration and Test (SSI&T) Preparation and Setup

26.1.3.1 Key Operator Roles

Science Coordinator: Provide support to Instrument Teams for the integration and testing of science software in the ECS system at the DAAC. Perform standard checking on all delivered software including source code, scripts, process control files and related documentation.

Science Data Specialist: Serves as a point-of-contact for planning, integrating, testing, and operating science software.

CM Administrator: Record, report, manage and distribute new and updated science software.

Science Software I & T Support Engineer: Provide support to Instrument Teams for the development, integration, test and problem resolution of science software.

Production Planner: Populate, maintain and schedule the production planning database for science software.

MODIS Science Data Processing Software Version 2.0 System Description Manual

This manual should be referred to for more detailed information on how to perform the SSI&T operational procedures as they apply to MODIS PGEs. It covers the specific attributes for each individual PGE and setup criteria.

26.1.4 COTS Software Tools

ClearCase: This tool is used as the ECS software configuration management tool. ClearCase provides a mountable file system, which is used to store version-controlled data, such as source files, binary files, object libraries and spreadsheets.

Distributed Defect Tracking System (DDTS): This tool is used to electronically process configuration change requests (CCRs). DDTS will prompt the user for relevant information, identify the request and will mail these requests to pre-designated personnel.

26.1.5 General Process

The SSI&T process consists of two activities:

- **Pre-SSI&T Activity** - During this activity the Delivered Algorithm Package (DAP) is inspected, and tested in a non-production environment.
- **Formal SSI&T Activity** - During this activity, the Product Generation Executives (PGEs) are integrated with the DAAC version of the SDP Toolkit and executed on the ECS PDPS platform.

Key Terms:

- **Product Generation Executives (PGEs)** - The smallest scheduled unit of science software.
- **Delivered Algorithm Package (DAP)** - An ensemble of PGE source code, makefile, documentation, and other related files delivered in a package from the SCF to the DAAC for SSI&T.
- **Process Control File (PCF)** - Relate logical identifiers to physical files and other parameters required by the PGE.
- **Strings -** The processing hardware on which the science software runs.
- **Archive** - A File Storage Type indicating that granules that will be inserted Data Server are intended for long term storage and acquisition for distribution.
- **Collection** - A related group of data granules.
- **Granule** - The smallest data element, which is, identified in the inventory tables.
- **Product** - A set of output values generated by a single execution of a PGE for archival by ECS. A PGE may generate one or more products whose attributes are defined by the data provider.
- **Reliability** - Software reliability means that the software runs to normal completion repeatedly over the normal range of data inputs and running conditions.

- **Safety** - Software safety means that the software executes without interfering with other software or operations.

The science software in the DAPs will be integrated onto the PDPS and be used to produce the output data as determined by the algorithms. The refined and updated DAPs and data produced by the science software will eventually be provided to the subscribing user. Before the PGE is integrated into a production environment, extensive testing on the software must be performed.

The following list provides a suggested, logical “road map” for getting science software tested and integrated into the ECS. This list is not intended to cover every situation and variations may be required.

26.1.5.1 General

- Science Software Integration and Test (SSI&T) is the process by which the science software is tested for production readiness in the DAACs in order to assure its (1) reliability and (2) safety. Prior to the delivery of the ECS software to the DAACs, SSI&T Checkout is conducted on early versions of the Products Generation Executives (PGEs) using separate system modes in the ECS PVC, VATC (Verification and Acceptance Test Configuration), or the DAAC environments.

- SSI&T activities can be broadly separated into two categories: pre-SSI&T and formal SSI&T. Pre-SSI&T activities are those which do not involve the ECS Planning and Data Processing (PDPS) or the Science Data Server (SDSRV), but the formal SSI&T activities do involve the full ECS including the PDPS and the SDSRV.

- Most steps in the SSI&T process are inter-related and some steps may assume that another step has been completed. The ordering of the steps is very important but it cannot, however be interpreted as a detailed, step-by-step guide to SSI&T activities.

- Science Software Integration and Test consists of the following activities most of which are fully detailed in Science Software Integration & Test Operational Procedures for the ECS Project (162-TD-001).

26.1.5.2 Pre-SSI&T Activities

1 As the DAP is delivered to a DAAC by the Instrument Team for SSI&T, the PGE listing documentation is reviewed.

2 The DAP is acquired and unpack and the documentation (i.e., packing list, readme, etc.) checked. The DAP contents are further checked by the Science Data Specialist to verify that the contents match the packing list, agreed-upon directory structures are employed, location of files are correct, and all intended files and directories are present.

3 The Science Data Specialist requests that the CM Administrator place the DAP under Configuration Management control using ClearCase.

4 The SSI&T team checks the science software for standards compliance using the Process Control File Checker to check process control files (PCF), and the Prohibited Function Checker to check source files. Extract and check prologs.

5 The SSI&T team builds the science software into PGEs using the SCF version of the SDP Toolkit. Compile all source code. Link object code with appropriate libraries. . If the SMF files compile successfully, then proceed to Step 11 below; otherwise, the problem needs to be fixed and a successful compile must occur before proceeding further. This may require one or more of the following:

6 Note any error messages and review the included documentation to ensure a proper compile;

7 Check environmental variables;

8 Review the setup files for proper directories and variables;

9 Make corrections and recompile.

10 If the executable builds successfully, proceed to Step 12. If the build fails, it may necessary to do one or more of the following before proceeding:

11 Check environmental variables;

12 Make corrections and repeat the build.

13 Run the PGE from the Command Line.

14 If it the execution is successful, then the output files (products) are checked using the SSIT Manager file comparison tools; otherwise, one or more of the following needs to be done before proceeding:

15 Check error logs;

16 Check environmental variables;

17 Review and source the setup files;

18 If necessary files are missing, then review the PCF file to ensure the existence of correct reference directories.

19 The SSI&T team runs and profiles the PGEs from the UNIX command line on the SGI, saving the profiling results. They will be used later when entering operational metadata into the PDPS.

20 The SSI&T team collects performance statistics for the PGEs.

21 The SSI&T team examines the output log files from the PGE runs for any anomalous message. The SSI&T team compares the output product data with the delivered test data using the file comparison tools. If the products do not match the delivered test outputs (expected outcome), the outputs should be analyzed and the PGE must be re-run. If the products match the delivered test outputs then

22 Steps 10 through 13 are repeated once using the DAAC Toolkit. If the products generated with the DAAC Toolkit match the delivered test output, formal SSI&T may begin.

23 SSI&T team reports any science software problems using the DDTS NCR process.

24 The SSI&T team reports any ECS problems using the DDTS NCR process.

25 The SSI&T team collects and logs all lessons learned.

26.1.5.3 Formal SSI&T Activities

1 For each ESDT used by the PGE, construct an ESDT ODL file for updating the PDPS or verify that they already exist. ESDT ODL files are also needed for all input and output data granules.

2 Construct a PGE ODL file for updating the PDPS database. This involves using the delivery PCF to construct an initial PGE ODL template file, which must then be hand edited to add required metadata. A mapping between logical IDs in the PCF and ESDT ShortNames must be known before this step is done.

3 Install ESDTs on the Science Data Server if verification indicates that they do not already exist. Installation links the PGE to all input and output ESDTs, which allows the PGE to run within the PDPS. **Note:** While installing ESDTs the SDSRV intermittently coredumps. To clean-up you must remove the ESDT from SBSRV and DDICT and then try again.

4 The Science Metadata.met is updated (PGE & ESDT Object Description Language or ODLs are created). This supplies metadata to the PDPS database

- If the Science Metadata update is successful, then the Operational Metadata is updated; otherwise, the ESDT ODL files may have to be checked for correctness before updating the Operational Metadata.

5 Register the PGEs with associated data in the PDPS database. This step uses the PGE ODL from step 2 above.

6 For each input dynamic data granule needed by the PGE, construct a Target MCF and insert both granule and .met files into the Science Data Server.

7 For each input static granule needed by the PGE, construct a Target MCF.met and insert both into the Science Data Server.

8 Assemble the SSEP (as a tar file) and Insert it to the Science Data Server.

9 Initiate a Production Request (PR) that will result in one or more DPRs.

10 Use the Planning Workbench to plan the PR and hence, run the PGE.

11 Monitor the PGE run using AutoSys. The PGE's progress is monitored using the AutoSys COTS. The distinct steps that are visible on the AutoSys GUI and whose success is evident are Resource Allocation (.Al), Staging (.St), Pre-Processing (.Pr), Execution of the PGE (.EX), Post-processing (.Ps), De-staging (.Ds), and De-Allocation of resources (.Da).

12 If any of the steps in the execution is not successful, then each failure must be identified and corrected before proceeding to the next step.

13 Examine the output Production History File from the PGE runs for any anomalous messages. Compare the output product data with the delivered test data using the file comparison tools. . If any of the steps in the execution is not successful, then each failure must be identified and corrected before proceeding to the next step.

14 If the output files match the test output files and they are in Hierarchical Data Format (HDF), they are visualized using the EOSView tool, or the Interactive Display Language (IDL) tool. If the files are not HDF, then IDL is used.

15 Using the Planning subsystem, initiate more complex Production Requests if chaining is required.

16 Using electronic or hard media transfer methods, distribute the data products to the Instrument Teams for their review.

Records

A weekly SSI&T status report is provided to NASA. This report contains the Performance Measurement Data.

Performance Measurements

SSI&T PGEs planned vs. actually delivered, pre-tested, and integrated is the metric used to monitor the effectiveness of the process described in the Procedure. Additionally, the Duration of Effort Required to Integrate in Work Days is used.

26.1.6 Preparation and Setup to Use the SSI&T Manager Tool

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

2 Enter the **password** then press the **Enter** key.

- Prior to the remote login, enter **setenv DISPLAY <local workstation IP address>:0.0** where the local workstation IP address represents the IP address you where you are located**.**
- You may need to setup the terminal so that the remote host is displayed on your screen. (Sun machine) This is done by clicking on the **Application Manager** icon (the file drawer located at the bottom of the screen), followed by the **Desktop Tools** icon**,** followed by the **Terminal Console** icon

3 Perform a remote login by typing **rlogin [host]** then press the **Enter** key.

- The **Enter Password** prompt is displayed.

4 Enter the **password** then press the **Enter** key.

5 Enter the directory where the setup script is located by typing **cd [**directory name**]** then press the **Enter** key.

6 Source the setup script by typing **source [**script name**]** then press the **Enter** key.

- The setup script contains directory paths, sets of alias commands, and tools for SSI&T.
- For example, source the SSI&T script: Type **source /usr/ecs/{MODE}/CUSTOM/utilities /.buildrc <RETURN>**
  Note: This step only needs to be done once per login.
- source .buildrc may not be supported on a particular software drop. Therefore the SSI&T scripts will be built into other another script.

7 To ensure access to the multi server environment when needed, the following generic login commands have been established and should be used routinely:

- From a terminal: xterm -n (host) &
- From the xterm invoked: **telnet (host)**
- login ID
- pw:
- setenv DISPLAY …..:0.0

8 Listed are some of the GUI tools, typical servers (examples and their Host that need to be considered for activation when conducting SSI&T:

- ECS Assistant, DM/, p0ins02,
- ECS Assistant, SDSRV/DSS, p0acs03
- ECS Assistant, DPS, p0sps06
- ECS Assistant, SBSRV/CSS, p0ins01
- SSIT Manager tools, AITTL/DPS t1ais01
- Production Request, PLS, p0pls01
- Planning Workbench, PLS, p0pls01
  Note: NETSCAPE should be closed to allow for a full screen GUI to be activated.
- Monitor PGE, p0pls01

9 A second xterm should be activated with the same login procedures so as to monitor the (log files) when entering SSI&T files from GUI**'s.**

10 Servers can be brought down in any order. To bring them backup requires that they be brought up in **a sequential order to ensure connectivity**, the order is listed as follows:

- **STMGT, MSS, DDIST, SDSRV, PDPS**

11 The above servers have unique hosts assigned. Each host needs to be logged into the **generic login: ID, pw:, and then press Enter Key.**

26.1.7 SSIT Software Operating Instructions

Starting the SSIT Manager GUI:

- On workstation **x0ais##**, at the UNIX prompt in a terminal window, type as in step 1 below your user id and password.
- NOTE: The **x** in the workstation name will be a letter designating your site:
- **g** = GSFC, **m** = SMC, **l** = LaRC, **e** = EDC, **n** = NSIDC, **o** = ORNL, **a** = ASF, **j** = JPL, **p**=PVC; the ## will be an identifying two-digit number (e.g., **g0ais01** indicates a Data Processing Subsystem (DPS) workstation at GSFC).
- Prior to the rlogin, enter **setenv DISPLAY <local_workstation IP address>:0.0.** The **<ipaddress>** is the ip address of **x0ais##,** and **xterm** is required when entering this command on a Sun terminal.

What the user must do before trying SSIT functionality:

1 **Example:** Log into an Algorithm and Test Tools (AITTL) environment using a machine so configured. At the PVC this machine is **p0ais01.** A special host has been established using the **id**: and **password**:. Type: **setenv DISPLAY ….:0.0**

2 **setenv <mode>** : **(cd /usr/ecs/<MODE>/CUSTOM/utilities** Note that this only has to be done once per login.

3 This directory should contain scripts pertaining to setting the environment for SSIT Manager. Type in: **EcDpAtMgrStart <mode> &**

- This invokes the **SSIT Manager GUI**, which should be displayed.

What must be done via SSIT tools:

Since SSIT is just a calibration of various tools, there is no specific order for which they must be run. Most tools can be brought up from the SSIT Manager GUI as well as started on their own.

The <u>File menu</u> provides the capability to exit the manager. The <u>Tools menu</u> provides access to the various tools that make up SSIT. The <u>Run menu</u> is customizable (allowing you to add your own scripts and tools) by editing the file *ssit_run_menu* in the *data/DPS* directory.

The <u>checklist</u> (first window on the GUI) allows you to check off various activities by double clicking on them. You may enter a commentary on the activity in the second window when checking off a particular item. The file checklist.sample in the data/DPS directory can be edited to change the items in the checklist or it’s location.

26.1.8 Updating the Leap Seconds and the Earth Motions Files

The toolkit requires Leap Second and Earth Motion updates, weekly and twice weekly respectively, to accurately compute most time conversions. The following scripts have been established to accomplish these tasks as part of ECS support.

update_leapsec.sh

This script updates the leapsec.dat file by ftp-ing to USNO and reformatting the information into the leap seconds file: $PGSHOME/database/common/TD/leapsec.dat

The present script, after obtaining the required file “tai-utc.dat” in the same Series 7 mentioned above, invokes PGS_TD_NewLeap, a C program that performs the actual update work. The function puts the current date in the header of the new leapsec.dat, with a remark that the file was either "Checked" (no new leap second) or "Updated" (new leap second). The date at which the USNO file used in the updating process was put on their server is also listed in the header.

update_utcpole.sh

This script updates the **utcpole.dat** file on the basis of new data obtained by ftp to the U.S. Naval Observatory in Washington, D.C (USNO). Their data file is excerpted and the required fields are reformatted and written into the utcpole file: $PGSHOME/database/common/CSC/utcpole.dat

The Leap Seconds file:

leapsec - file ID: **$PGSHOME/database/common/TD/leapsec.dat**

(Atomic time from International Earth Rotation Service)

Introduced every 12 to 24 months, announced almost 6 months in advance or as little as 90 days notice. Update available from U.S Navy Observatory (USNO).

Interval of update recommend: weekly, except Sundays 17:45 hours to 17:55 Eastern US time. Runtime is approximately 30 seconds**.**

The Earth Motion file:

utcpole – file ID: **$PGSHOME/database/common/CSC/utcpole.dat**

(Record of the Earth’s variable of slowing rotation with respect to UTC Time.) **Interval of update recommended: Twice weekly except Sundays 17:45 hours to 17:55 Eastern US time. Recommended scripts be run in the afternoon or evening each Tuesday and Thursday.**

26.1.9 Script Name: update_leapsec.sh

The following processing tasks are carried out automatically by the use of this script:

- **Update via: Ftp to USNO, “maia.usno.navy.mil”** file accessed for leapsec: **tai-utc.dat. (Tests connectivity by using “Whazzup")**
- **Function to be applied: PGS_TD_NewLeap,** excerpts and reformats the new information and appends new data and date to **leapsec.dat** file. A remark that the file was either “Checked” (no new Leap second) or “Updated” (new leap second). 26.2.2 Script Name: update_utcpole.sh

The following processing tasks are carried out automatically by the use of this script

- **Update via: Ftp to USNO, “maia.usno.navy.mil”** file accessed for utcpole: **finals.data. (Tests connectivity by using “Whazzup")**
- **Function to be applied: PGS_CSC_UT1_update,** excerpts and reformats the new information and appends new data to **utcpole.dat** file.

Guidelines:

1 The script must be run on a machine that has the Toolkit mounted and which can access the USNO site via ftp and access e-mail. (p0spg01 used at the Performance Verification Center (PVC))

2 For each installed Toolkit (including all modes, such as debug, F77, F90, etc.) the scripts need to be run only once, even if different platforms or operating systems are run. However, if entirely separate Toolkits exist at your installation, with different $PGSHOME home directories, then either the scripts need to be run in each, or the data files can be propagated from a primary Toolkit to the others.

3 It is highly desirable to have outgoing e-mail mounted on the machine of choice, so that error messages may be issued automatically from the scripts in case of failure.

4 If the updating process fails, then the script must be rerun. The Toolkit team should be contacted anytime the scripts are not giving the correct or accurate. It is highly desirable to have outgoing e-mail mounted on the machine of choice, information. The 2 sets of scripts do also send an email message to SDP Toolkit mail address when a script fails

5 The Toolkit requires that the two data files not be too stale. Therefore the useful lifetime of the utcpole.dat and leapsec.dat files is 83 days. The Toolkit will issue an error message if no update was performed beyond 83 days. If this occurs you can expect geolocation accuracy to deteriorate to an extent that could require re-running for some of the more stringent users. If Toolkit requires a leap second value after this date, an error message will be returned. This generally means that production will cease.

6 Keep the Latest files until your updates are completed! They are useful for a backup should they be needed.

Hardware Needed and Setup Procedures

The user's environment needs to be set up by running the script $PGSBIN/pgs-dev-env.csh or $PGSBIN/pgs-dev-env.ksh, depending on the shell being used. $PGSBIN stands for $PGSHOME/bin/mach, where "mach" stands for one of: sun5, sgi64, sgi, sgi32, ibm, dec, or hp. In other words it is shorthand for the machine "flavor" you are using, and for sgi, the compiler option. Not all versions are necessarily at each DAAC or SCF, and in some cases the path may be more complicated. For example, at Goddard Space Flight Center DAAC, typical binary directories are /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit/bin/sgi64_daac_f77/, or /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit/bin/sgi64_daac_f90_debug/, for example.

Once the setup script is located and sourced, $PGSBIN is defined and your path includes it. Furthermore, a "PCF", or process control file, $PGS_PC_INFO_FILE is defined, which allows the executable functions invoked by the scripts to find the old data files, which are needed for the updates.

To run the scripts successfully, you must have write permission on the data files.

After the setup is done, just run the scripts. Both scripts (update_utcpole.sh and update_leapsec.sh) are located in the directory $PGSBIN, which will be in your path after the Setup script has been run.

- On workstation **x0spg##**, at the UNIX prompt in a terminal window, type **source /data3/ecs/TS1/CUSTOM/daac_toolkit_f90/TOOLKIT/bin/sgi64/pgs-dev-env.csh .** This will set up the various environment parameters, such as PGSHOME, to enable the 64 bit version of the FORTRAN 90 compiler to be run.
- **NOTE:** The **x** in the workstation name will be a letter designating your site:
- **g** = GSFC, **m** = SMC, **l** = LaRC, **e** = EDC, **n** = NSIDC, **o** = ORNL, **a** = ASF, **j** = JPL **p=**PVC; the ## will be an identifying two-digit number (e.g.,**g0spg03** indicates a Science Processor Subsystem workstation at GSFC).
- Prior to the rlogin, enter **setenv DISPLAY <local_workstation IP address>:0.0.** The **<ipaddress>** is the ip address of **x0spg**

Example: To Update the Latest Leapsec.dat and Utcpole.dat files perform the following steps:

1 telnet to a machine that supports the Toolkit. (**telnet p0spg01**)

2 login: **ID**, **Password:**

3 **setenv DISPLAY ….:0.0**

4 **setenv PGSHOME /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit**

5 **cd /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit/bin/sgi_daac_f77** then

6 **source pgs_dev-env.csh**

7 For **leapsec: cd /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit/database/common/TD**

8 **cp leapsec.dat leapsec.dat_old**

9 Know thread for Leap Second run: **cd /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit/src/TD** then do **ls –** select: update_leapsec.sh or run script for Leap Second type in: **update_leapsec.sh**

- A successful update will look like the following:
  p0spg01{cmops}[288]->update_leapsec.sh
  Status of PGS_TD_NewLeap call was (0)
  Status of MOVE command was (0)

For utcpole:

1 **cd /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit/database/common/CSC**

2 **utcpole.dat utcpole.dat_old**

3 Know thread for utcpole run: **cd /usr/ecs/OPS/CUSTOM/TOOLKIT/toolkit/src/CSC** then do ls – select: **update_utcpole.sh** or run script for utcpole type in: **update_utcpole.sh**

- A successful update will look like the following:
  p0spg01{cmops}[294]->update_utcpole.sh
  Status of PGS_CSC_UT1_update call was (0)
  Status of MOVE command was (0)

26.2 Science Software Integration and Test (SSIT) Manager

26.2.1 SSIT Manager Overview

The principal tool used during SSI&T is the SSIT Manager. The SSIT Manager is the top-level graphical user interface (GUI) environment presented to SSI&T personnel. Its purpose is to bring together the tools needed for SSI&T into a single, graphical environment. Please refer to Operational Tools Manual for the ECS Project 609-EMD-001, Section 4.5 for the latest update on expanded uses of the SSIT Manager.

The SSIT Manager is the starting point for use by the SSI&T specialist to check in and verify the science software delivered by the instrument teams at the SCFs. The SSIT Manager application provides access to all COTS tools and custom applications that are part of the SSI&T environment. The SSIT Manager GUI is capable of kicking off instrument-specific compilation and execution scripts, configuration management scripts, custom code checking, file display and comparison tools, and COTS tools such as analysis environment programs. The SSIT Manager GUI contains a checklist of SSI&T steps in delivery and testing of science software, and a display of a log file recording SSI&T events. The checklist and log are the only inherent functionality to SSIT Manager; all other programs run from the Manager are also accessible from the Unix command line.

The terms Process Control File (PCF) and Object Description Language (ODL) are used in the following sections. The PCF is a file that tells an executable where to find its various inputs and outputs, as well as the values for any specific runtime parameters. Different versions of these files are used both by the SSIT Manager and by PGEs. ODL is a parameter = value format for input files. It is used to define the PGEs to the Planning and Data Processing Database.

Table 26.1-1 presents a summary of the capabilities provided via the SSIT Manager GUI.

Table 26.1-1. Common ECS Operator Functions Performed through the SSIT Manager GUI (1 of 3)

Operating Function	Command/Script or GUI	Description	When and Why to Use
Prepare SSI&T checklist.	SSIT Manager GUI	• SSIT Manager GUI requires a checklist which it can access • the checklist contains all the operational procedures for science software integration and test	• during normal SSIT operations, used to keep track of activities pending and completed • checklist of operational procedures must first be prepared using a text editor. • SSIT Manager must then be linked to the checklist before invoked
Change SSI&T Checklist.	SSIT Manager GUI	• SSIT Manager GUI requires to change the checklist • the checklist contains all the operational procedures for science software integration and test	• Update tracking of activities pending and completed • checklist of operational procedures must first be prepared using a text editor. • SSIT Manager must then be linked to the checklist before invoked
open xterm session.	• open xterm window via Tools pull-down menu • also can be opened via Unix command *xterm*	Standard Unix command line window.	As needed for ad hoc use.
Code analysis.	select SPARCworks via Tools: Code Analysis pull-down menu	Used for ad hoc analysis of science software.	used to debug problems (e.g., memory leaks)
check for standards compliance.	select the following via Tools: Standards Checkers pull-down menu: • FORCHECK • Prohibited Function Checker • Process Control File (PCF) Checker • Prolog extractor	• check FORTRAN 77 science software • check if certain functions are used in the science software which conflict with the production environment • check the syntax of the data in the Process Control File • Extract prologs from science software	• to ensure that science code conforms to ECS standards • to ensure that the delivered PCF is of the proper syntax • To extract prologs from science software

Table 26.1-1. Common ECS Operator Functions Performed through the SSIT Manager GUI (2 of 3)

Operating Function	Command/Script or GUI	Description	When and Why to Use
Product Examination.	select the following via Tools:Product Examination pull-down menu: • IDL • EOSView	Opened via Tools pull-down menu.	• Ad-hoc graphical analysis • For viewing an arbitrary file (e.g., standard product) in HDF format
File Comparison.	select the following via Tools: Product Examination File Comparison • ASCII files • Binary files • HDF files (GUI) • HDF files (hdiff)	Compares the outputs of the science software between the DAAC and SCF.	Ensures that the output that was generated at the SCF when running the science software is the same output that is generated at the DAAC.
Edit text.	select the following via Tools: Text Editors pull-down menu: • Emacs • Xedit	Text editors.	Edit arbitrary text file.
PDPS Database	select the following via Tools:PDPS Database pull-down menu: • PCF ODL Template • Check ODL Files • SSIT Science Metadata Update • SSIT Operational Metadata Update GUI	• creates an ODL file template from the science software PCF • Check the ODL file updates PGE and ESDT SCIENCE metadata in the PDPS /SSIT database • updates PGE OPERATIONAL metadata via GUI in the PDPS /SSIT database	To initialize and update the Planning/Production (PDPS) databases: • SSIT version • Production version

Table 26.1-1. Common ECS Operator Functions Performed through the SSIT Manager GUI (3 of 3)

Operating Function	Command/Script or GUI	Description	When and Why to Use
Data Server	select the following via Tools: Data Server pull-down menu: 1. Acquire DAP 2. Insert Static 3. Insert Test Dynamic 4. Insert EXE TAR 5. Edit SSAP 6. Get MCF	1. Acquires DAP. 2. Inserts static input file. 3. Inserts test dynamic input file. 4. Inserts tar file with files needed for Processing. 5. Edits Science Software Archive Package (SSAP) components. 6. Acquires Metadata Configuration Files from the Data Server.	1. After DAP notification received by email. 2. After ESDT is registered in Data Server, before test PGE run. 3. After ESDT is registered in Data Server, before test PGE run. 4. After PGE compilation, before test PGE run. 5. After PGE testing is complete, at time of promotion to Production, as needed to edit/review SSAP components. 6. As needed to retrieve MCF.

26.2.2 Quick Start Using SSIT Manager

The SSIT Manager provides a common interface to the SSI&T tools. An overview of the SSIT Manager GUI is provided in Section 26.1.2. A more detailed discussion of the tools accessed via this GUI is provided in subsequent sections.

The following assumptions are made with regard to the use of the SSIT Manager application.

1. The operator is located at a workstation or server to which the SSIT Manager has been configured.
2. The operator has proper authorization to access the PDPS/SSIT database and the Data Server.
3. To access files in ClearCase, the operator has a ClearCase view already set.
4. The operator's environment has been configured as documented in the pertinent sections of the Help menu, available from the main window of the SSIT Manager. The Index submenu of the Help menu provides access, through the Netscape browser, to a number of topics that help the operator in the environment configuration. The Help menu contains a list of topics that can be searched through.

26.2.2.1 Invoking SSIT Manager from the Command Line Interface

To start SSIT Manager at the Unix command line:

> ***DpAtMgr ConfigFile* <config_filename> ecs_mode <MODE> &**

where **config_filename** is the name of the user's personal Process Framework configuration file for this program, as customized from **$ECS_HOME/CUSTOM/cfg/**.

ecs_mode is the ECS mode of operation, e.g., **OPS, TS1**

26.2.2.1.1 Sun Platform

Table 26.1-2 lists the SSI&T command line interfaces for the Sun workstation.

Table 26.1-2. Command Line Interfaces (Sun) (1 of 2)

Command Line Interface	Description and Format	When and Why Used
EcDpAtMgr	Startup of SSIT Manager.	To do SSI&T, and record items accomplished in the log.
EcDpAtMgrLogDump	Used to dump/print a log file to the screen.	As needed.
Xterm	Open a Unix command line window.	As needed.
sparcworks	Ad hoc code analysis	As needed.
ghostview	Postscript file viewer	As needed.
Netscape	WWW browser Netscape	As needed.
Acroread	PDF file viewer Adobe Acrobat	As needed.
DpAtMgrForcheck	FORTRAN 77 code analysis	Determine whether FORTRAN 77 science software adheres to standards.
EcDpAtBadFuncGui	Prohibited function checker (GUI)	Determine whether science software adheres to standards.
EcDpAtBadFunc	Prohibited function checker (command line)	Determine whether science software adheres to standards.
EcDpAtCheckPCF	Process Control File checker (GUI)	Determine whether PCF is valid.
EcDpAtMgrPrologs	Prolog extractor	Extract science software code prologs.
/data/IDL/idl_4/bin/idl	IDL	As needed.
EOSView	EOSView	HDF file viewer.
EcDpAtMgrXdiff	ASCII file comparison	Compare 2 text files.
EcDpAtBinDiffGui	Binary file difference environment	Compare 2 binary files.
DpAtCheckHdfFile	HDF file comparison (GUI)	Compare 2 HDF files.
DpAtHdiff	HDF file comparison (command line)	Compare 2 HDF files.
Xedit	Text editor	As needed.
Emacs	Text editor	As needed.
EcDpAt CreateODLtmplate	Create PGE metadata ODL template file.	Before running EcDpAtDefinePGE .
EcDpAtVerifyODL	Verify PGE metadata ODL template file.	Before Running EcDpAtDefinePGE.

Table 26.1-2. Command Line Interfaces (Sun) (2 of 2)

Command Line Interface	Description and Format	When and Why Used
EcDpAtDefinePGE	Update PDPS/SSIT database with SCIENCE metadata.	Before executing PGE in SSIT environment.
DpAtOpDbGui	Update PDPS/SSIT database with OPERATIONAL metadata.	Before executing PGE in SSIT environment.
EcDpAtStageDAP	Acquires DAP from the Data Server.	After email subscription notification received.
DpAtInsertStatic	Inserts static input file into the Data Server.	Before executing PGE in SSIT or Production environment.
DpAtInsertTest	Inserts test dynamic input file into the Data Server.	Before executing PGE in SSIT environment.
DpAtInsertExeTar	Inserts tar file of executables, etc. needed to run PGE file into the Data Server.	Before executing PGE in SSIT or Production environment.
EcDpAtSSAPGui	Edit and inserts a single SSAP component into the Data Server.	After SSI&T is finished, before official promotion to Production.
netscape <html page name>	HTML pages for acquiring SSAP components from the Data Server, including test outputs.	During SSI&T, to get test outputs; After SSI&T is finished.
EcDpAtaCQUIREMCF	Get ESDT from the Data Server and insert MCF.	Before inserting MCF in the Data Server.

26.2.2.1.2 SGI Platform

It is intended that the SSI&T tools be most often run from the SSIT Manager. A small number of SSIT tools run only on the SGI platform. Because of security considerations, these tools cannot be run from the SSIT Manager on the Sun. They may only be run from the Unix command line on the SGI platform as indicated in Table 26.1-3.

Table 26.1-3. Command Line Interfaces (SGI)

Command Line Interface	Description and Format	When and Why Used
usr/sbin/cvproj	ProDev Workshop: Used for ad hoc analysis of science software.	Used to determine causes of problems (e.g., memory leaks).
DpAtRusage	Measures PGE performance.	Output of this tool is to be typed into the “Performance Statistics” section of the PROFILE screen of the PDPS/SSIT Database Update GUI.

Table 26.1-4 lists SGI platform tools associated with the SSIT process.

Table 26.1-4. SGI Tools Description

Categories/Tools	Tool Description & Use	Further Information
ProDev Workshop	• ProDev Workshop is a COTS package developed by SGI • This tool is targeted within ECS for applications running on the SGI science processors • ProDev Workshop is a software development support tool which includes several tools that may have applicability to SSIT • Among these tools is the capability to perform static code analysis to aid in the detection of memory leaks	• ProDev Workshop includes online documentation describing its features • Other ProDev Workshop documentation is delivered with ECS. • ProDev Workshop is not available from the SSIT menu. This tool must be started from the Command Line Interface: see Table 4.5.1-3.
PGE Performance	• DpAtRusage is a custom tool developed by ECS • It measures performance parameters such as CPU time used for a PGE linked to the SDP Toolkit, SCF version.	A help message is printed if the tool is invoked without input parameters.

Across the top of the SSIT Manager GUI are the toolbar items **File**, **Tools**, and **Run**. Clicking on each of these invokes a pull-down menu.

Under the **File** pull-down menu, the only item is **Exit**. Clicking on this causes the SSIT Manager to terminate.

The **Tools** pull-down menu has most of the SSIT Manager's tools. The menu functions are:

The **Run** pull-down menu initially contains no menu items. Its purpose, however, is to allow a place for SSI&T personnel to place their own custom tools and scripts.

26.2.3 SSIT Manager GUI

The SSIT Manager GUI is the starting point for SSI&T activities. It provides access to a collection of tools that will be useful for this purpose.

26.2.3.1 General Setup of the SSIT Manager

The SSIT Manager requires a configured environment within which to run; it runs only on the AIT Suns. The set up steps described in this section need only be done the first time a SSI&T operator uses the SSIT Manager

To set up the environment for the SSIT Manager, execute the procedure steps that follow.
(This procedure was tested by **telnet p0ais01**, ID: **___**, PW:**___, setenv DISPLAY** ***Example:***
155.157.123.34:0.0 or **setenv DISPLAY p0ais01:0.0** .

1 **setenv ECS_HOME /usr/ecs & setenv <mode>**

2 **cp /usr/ecs/*mode*/CUSTOM/data/DPS/DpAtMgrInternal.pcf $HOME/*mySSITpcf***, press **Return**.

- The ***mode*** is the ECS mode in which you are operating. This mode should be **TS1.**
- The ***mySSITpcf*** is the file name of the private copy of the PCF that the SSI&T operator will use when running the SSIT Manager. The **$HOME** is the environment variable for the user's home directory. For example, **cp /usr/ecs/TS1/CUSTOM/data/DPS/DpAtMgrInternal.pcf $HOME/myPCF**, press **Return**.

3 At the UNIX prompt on the AIT Sun, type **setenv PGS_PC_INFO_FILE $HOME/*mySSITpcf***, press **Return**. (Check **env** for proper home path)

- The ***mySSITpcf*** is the full path name to the private copy of the PCF to be used with the SSIT Manager when you run it (from step 1).
- It may be useful to add this line to your **.cshrc** (or other start up script) so that it is set every time you login.

4 At the UNIX prompt on the AIT Sun, type **cd /usr/ecs/*mode*/CUSTOM/utilities**, press **Return**.

- The ***mode*** is the ECS mode in which you are operating. This mode should be **TS1 or another mode assigned beforehand to operate in.**

5 At the UNIX prompt on the AIT Sun, type **EcDpAtMgrStart <mode> &**

- This invokes the **SSIT Manager GUI**, which should be displayed.
- The checklist displayed within the GUI will be the default.
- This sets environment variables and other settings needed for running the SSIT Manager.

26.2.3.2 Setup of a Checklist for the SSIT Manager

The SSIT Manager offers the capability of maintaining user-defined checklist of SSI&T activities. The checklist is presented in the main window of the SSIT Manager. A default checklist is displayed unless a new checklist is specifically created. This procedure explains how to set up a customized checklist.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Creating a User-Defined Checklist for the SSIT Manager:

1 a From the SSIT Manager, click on the **Tools** menu, then choose **Product Examination**, then **EOSView**.

- The EOSView GUI will be displayed.

1 b Alternately, if EOSView isn't available from the SSIT Manager GUI, invoke EOSView from the command-line.
Go to the proper area by typing **cd /usr/ecs/TS1/CUSTOM/eosview** <**RETURN**>
Start EOSView by typing **EOSView** <**RETURN**>

2 In the GUI labeled **EOSView - EOSView Main Window**, click on the **File** menu and select **Open.**

- The **Filter** GUI will be displayed.

3 In the subwindow labeled **Filter**, enter full path name and file name wildcard template. For example, enter ***/home/MyDirectory/MySubdirectory/****.

- The ***/home/MyDirectory/MySubdirectory/**** represents the location to the directory containing the HDF-EOS files to examine.
- The asterisk **(*)** is a wildcard template that represents all files in that directory; other wildcard templates can narrow the search further, *e.g.* ***.hdf**.
- Use the **Directories** field to further select the correct directory.
- Files found matching the wildcard template in the chosen directory will be displayed in **Files** subwindow.

4 In the **Files** subwindow, click on the file name of the HDF-EOS file to examine. Then click on the **OK** button.

- A GUI labeled **EOSView - *MyOutputFile.hdf*** will be displayed where ***MyOutputFile.hdf*** is the file name of the file chosen in step 3.
- Be patient - this GUI may take some time to appear, particularly for large files.
- Once displayed, a list of HDF objects will appear in the main window. If nothing is listed, it means that no HDF objects were found within the file.

5 In the GUI labeled **EOSView - *MyOutputFile.hdf***, click on an object listed for which metadata is to be inspected.

- The object selected will be highlighted.
- Do not double click on object since this will cause a **Dimension** GUI to be displayed instead.

6 In the GUI labeled **EOSView - *MyOutputFile.hdf***, click on the **Attributes** menu and select **Global**.

- A GUI labeled **EOSView - Text Display** will be displayed.
- The global metadata associated with the object selected (in step 5) will be displayed in a scrollable field.
- If instead, the message "Contains no Global Attributes" appears, then the selected object contains no global metadata.

7 Repeat steps 5 and 6 for each HDF object within the selected HDF-EOS file for which metadata is to be examined.

8 In the GUI labeled **EOSView -** ***MyOutputFile.hdf***, click on the **File** menu and select **Close**.

- The **EOSView -** ***MyOutputFile.hdf*** GUI will disappear.
- Be patient - this GUI may take some time to disappear, particularly for large files.

9 In the GUI labeled **EOSView - EOSView Main Window**, click on the **File** menu and select **Exit**.

- The **EOSView - EOSView Main Window** GUI will disappear.

26.2.4 SSIT Manager Tools

There are several tools that are accessible through the SSIT Manager GUI. After selecting the TOOLS menu option of the menu bar, a set of options is available.

26.2.5 Using the SSIT Manager

The following is a list of tools and/or assumptions:

1. The SSIT Manager is running.
2. The source file(s) are available, accessible, and have read permissions.
3. The below listed formatted text (ASCII) files containing the list of prohibited functions exist in the directory stored in the environment variable DPATMGR_DAT:

   prohibitedFunctionsAda.txt
   prohibitedFunctions.C++.txt
   prohibitedFunctions.C.txt
   prohibitedFunctions.F77.txt
   prohibitedFunctions.F90.txt
4. If the source code files to be checked are in a VOB in ClearCase, a view has been set before the SSIT Manager was started.

26.3 Delivered Algorithm Package (DAP) - Insert/Acquire, Unpack, Subscription

The **insert** service is used to put the DAP into the Science Data Server. Once the DAP is in the Science Data Server, the **acquire** service is used to retrieve it.

The Delivered Algorithm Package (DAP) is the vehicle by which the PGE, source code, supporting files, documentation, etc. are delivered to a DAAC for SSI&T. Typically, the DAP is a compressed TAR file with a file name of form *string*.tar.Z . After initial processing, the DAP is broken apart into its components and those components will be subsequently processed and used based on their intended function.

The delivery mechanism for DAPs can be electronic (e.g. via UNIX ftp) or physical media (4 mm or 8 mm digital audio tapes).

26.3.1 Acquiring the Delivered Algorithm Package (DAP)

The following procedures are used by the SSIT team to acquire DAPs.

26.3.1.1 Acquiring the DAP via FTP

FTP is a method that the SSIT team uses in order to receive the science software. The following example demonstrates the FTP of the tar file from a remote machine.

Acquiring the DAP via FTP

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

2 Enter the **password** then press the **Enter** key.

3 At a UNIX prompt, type **cd *DeliveryPathname***, then press the **Enter** key.
- The ***DeliveryPathname*** is the full path name to the directory that has been set aside for ftp pull of DAPs from the Instrument Team. For example, **cd /home/*user*** where ***user*** is the user's login directory, then press the **Enter** key.
- If the DAP is to be copied into a subdirectory, change to this subdirectory.

4 At a UNIX prompt, type **ftp *machineIPaddress***, then press the **Enter** key.
- The ***machineIPaddress*** is the IP address or fully qualified domain name of the remote SCF machine. For example, **ftp 192.266.53.2**, then press the **Enter** key.
- Or for example, **ftp g0ais01.gsfc.ecs.nasa.gov,** then press the **Enter** key. The remote machine will likely display some messages and then prompt for a login name.
- An ftp session is established.

5 At the ftp prompt on the remote machine, enter user login name, then press the **Enter** key.
- The remote machine will typically respond with **331 Password required for *username*:**

6 At the ftp prompt on the remote machine, enter user password, then press the **Enter** key.
- The remote machine will typically respond with **230 User *username* logged in** and display the **ftp>** prompt for further ftp commands.

7 At the ftp prompt on the remote machine, type **cd *DAPpathname*** then press the **Enter** key.

- The ***DAPpathname*** is the full path name to the directory on the remote machine containing the DAP to retrieve. For example, **cd /home/mac ,** then press the **Enter** key. The directory location should be known.

8 At the ftp prompt on the remote machine, type **binary**, then press the **Enter** key.

- The **binary** command causes subsequent file transfers to be in binary mode, preserving the integrity of the file to retrieve without interpretation (as would be done in ASCII mode.
- The system will typically respond with the message **200 Type set to I** indicating that binary mode has been set.

9 At the ftp prompt on the remote machine, type **get *DAPfilename***, then press the **Enter** key.

- The ***DAPfilename*** is the file name of the DAP to retrieve.
- For example, type **get TestPGE.tar**, then press the **Enter** key.
- The user may need to type **dir** then press **Enter** to display a listing of the files in the current directory. The system will likely display several lines of messages once the transfer has completed. For large files, this may take a long time (minutes to hours depending upon the size of the DAP and the bandwidth of the connection).

10 At the ftp prompt on the remote machine, repeat step 9 or type **quit**, then press the **Enter** key.

- Typing **quit** and pressing **Enter** closes the ftp connection with the remote machine.
- Retrieve other DAP files by repeating step 9. The DAPs retrieved will reside in ***DeliveryPathname*** on the local machine.

11 At the UNIX prompt type **cp /home/mac/TestPGE.tar**, then press the **Enter** key.

- This step will copy the DAP tar file into their working directory.

26.3.1.2 Acquiring the DAP from the Archive after Ingest

The **Insert** service is used to put the DAP into the Science Data Server. After it is ingested, the **Acquire** service is used to retrieve it.

The DAP is acquired from Data Server and placed in the specified directory. Note there will be 2 files, the DAP itself (a big tar file) and the metadata associated with the DAP. The metadata may be helpful in the creating the SSAP.

When a DAP is inserted into the Data Server by Ingest, an email is sent to all users who subscribe to that event (Section 26.3.2).

26.3.1.3 Performing a DAP Acquire Using SSIT Manager

Generally, the preferred approach to accomplishing a DAP **Acquire** will be through the use of the SSIT Manager GUI.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The following servers/services are up and operational:

   Data Server, Subscription Server, Storage management

2. The following must have occurred between those Servers/Services:

   Ingest must have ingested DAP and Inserted it into the Data Server. Subscription Server must have gotten notification from the Data Server of the Insert.
   Subscription Server must send email to the SSIT operator notifying him/her of DAP Insertion and giving him (in the email) the UR of the DAP.

4. The SSIT Manager is available.

5. The X Window **DISPLAY** environment variable is pointing to your screen.

DAP Acquire Procedures:

1 If not already on an AIT Sun, log onto one from your current machine.

2 Bring up the SSIT Manager GUI. At the UNIX prompt, type mgr (if alias has been established)

3 After a short while, the SSIT Manager GUI will appear. From the SSIT Manager top menu bar, select **Tools -> Data Server -> Acquire DAP**
If the SSIT Manager GUI is used to initiate the DAP processing, Step 4 can be skipped.

Alternately, one can initiate the DPA processing sequence from the command line. To do this.

1 Type **source /usr/ecs/TS1/CUSTOM/bin/DPS/.buildrc <RETURN>**
Note: This step only needs to be done once per login

2 Type **/usr/ecs/TS1/CUSTOM/bin/DPS/DpAtStageAlgorithmPackage.sh <RETURN>**

- The user will be prompted with:
  ** DAP Staging Tool **
  Configuration filename? (enter for default: DpAtAA.CFG)

3 To respond, type **<RETURN>**

- The user will be prompted with:
  ECS Mode of operations? (enter for default: OPS)

4 To respond, type **TS1 <RETURN>**

- The user will be prompted with:
  Name of email message file (including path)?

5 To respond, type the required file name plus the path, e.g., **/home/diascone/emessage01.asc <RETURN>**

- The user will be prompted with:
  Directory to receive staged file?

6 To respond, type the required directory, e.g., **/home/diascone/staged <RETURN>**

26.3.2 Unpacking a DAP

Once a DAP has been acquired via electronic means or physical media, it typically needs to be unpacked before its contents are accessible for SSI&T. Several mechanisms are available under standard UNIX for packing and unpacking files to and from a file archive, the most common being UNIX *tar*. Another fairly typical utility is *gzip* and its companion, *gunzip*.

The file name extension is usually an indication of the packing utility used and DAP files should use this convention. DAP files that have been packed using the UNIX *tar* utility will usually have .tar as a file name extension indicating a tar file. If the DAP has been further compressed using the UNIX *compress* utility, the file name extension is typically .tar.Z indicating a compressed tar file. For DAP files packed with the *gzip* utility, the .zip file name extension is generally used.

When unpacking is performed on a DAP, the contents of the packed file are moved from the tar archive to local disk. If the DAP tar file contains directories as well as files, these directories will be created in the same structure as in the tar file. This structure typically reflects the directory structure from which the tar file was created in the first place at the SCF. Once a tar file has been unpacked, the original tar file will still exist unaltered.

Unpacking a DAP

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

2 Enter the **password** then press the **Enter** key.

3 At a UNIX prompt, type **cd UnpackPathname**, then press the **Enter** key.

- The ***UnpackPathname*** is the path name of the directory that has been set aside for unpacking of DAPs.

- This directory now contains the DAP tar file. For example, **cd /home/*user***, where ***user*** is the user's login directory, then press the **Enter** key.

4 If the tar file is compressed, at a UNIX prompt, type **uncompress *PackedDAP.Z***, then press the **Enter** key.
- The ***PackedDAP.Z*** is the file name of the compressed DAP file.
- The file name extension of **.Z** is a convention indicating UNIX compressed files. The **uncompress** utility expects this file name extension by default. A resulting error may indicate that the DAP file was not compressed or that another compression utility was used. If the file name extension was **.Z**, the uncompressed version will have the same file name but without the **.Z**, for example ***PackedDAP***.
- The tar file for the SSI&T Training will not be compressed.

5 At the UNIX prompt, type **tar xvf *PackedDAP***, then press the **Enter** key.
- The ***PackedDAP*** is the file name of the uncompressed DAP file.
- The tar archive will be unpacked in the current directory. If the archive contained directories and subdirectories, these will be created by the tar utility and populated by the files that belong.

26.3.3 Performing a DAP Insert

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The DAP ESDT has been installed on the Data Server.
2. The Target MCF for the DAP has been created for the Insert.
3. The SSIT Manager is running.

To Insert a DAP to the Science Data Server, execute the following steps:

1 From the SSIT Manager, click on the **Tools** menu, then choose **Data Server** and then **Insert Test Dynamic**.
- An xterm with title "SSIT: PGE Test Dynamic Input File Insertion" will be displayed.

2 At the program prompt **Configuration filename? (enter for default: ../..cfg/EcDpAtInsertTestFile.CFG)**, press **Return**.

3 At the program prompt **ECS Mode of operations?**
- Type in the **<mode>** you are working in. For example, **TS1** or **OPS.** Press **Return.**

4 At the program prompt **ESDT short name for the file(s) to insert?** type ***ESDTShortName***, press **Return**

- The ***ESDTShortName*** is the ShortName for the DAP file, which is DAP.

5 At the program prompt **ESDT Version for the file(s) to insert?** Type in the ESDT version and press **Return.**

6 At the program prompt **Is there more than one data file to this Dynamic Granule (Y = Yes, N = No)? (enter for default: N)?**

7 At the program prompt **Single Filename to Insert? (including FULL path)** type ***pathname*/*GranuleFilename***, press

- The ***pathname*/*GranuleFileName*** is the full path name and DAP file name.

8 At the program prompt **Associated ASCII Metadata Filename to Insert (including FULL path)** , Type ***pathname*/*DAP.tar.met*** and press **Return**.

- ***pathname*** is full name of the path and ***DAP.tar.met*** is the name of the associated .met file.

9 At the program prompt **Hit return to run again, 'q <return>' to quit:** type **q** and press **Return** to quit or just press **Return** to insert additional dynamic granules.

- If continuing, repeat steps 2 through 8.

26.3.4 An Example of a DAP Metadata File

The following is an example of a DAP target metadata file. Three attribute values needs to be modified based on each DAP. The attributes are DAPPGEName, DAPPGEVersion, and DAPSWVersion.

GROUP = INVENTORYMETADATA

```
GROUP            = INVENTORYMETADATA
  GROUPTYPE        = MASTERGROUP

  GROUP            = ECSDATAGRANULE

    OBJECT           = LOCALGRANULEID
      NUM_VAL          = 1
      VALUE          =
"MOD04L_O.A1997226.0704.002.1999204222146.hdf"
    END_OBJECT         = LOCALGRANULEID

    OBJECT           = PRODUCTIONDATETIME
      NUM_VAL          = 1
      VALUE          = "1999-07-23T22:21:46.000Z"
    END_OBJECT         = PRODUCTIONDATETIME
```

```
    OBJECT                 = DAYNIGHTFLAG
      NUM_VAL              = 1
      VALUE                = "Day"
    END_OBJECT             = DAYNIGHTFLAG

    OBJECT                 = REPROCESSINGACTUAL
      NUM_VAL              = 1
      VALUE                = "processed once"
    END_OBJECT             = REPROCESSINGACTUAL

    OBJECT                 = LOCALVERSIONID
      NUM_VAL              = 1
      VALUE                = "002"
    END_OBJECT             = LOCALVERSIONID

    OBJECT                 = REPROCESSINGPLANNED
      NUM_VAL              = 1
      VALUE                = "further update is anticipated"
    END_OBJECT             = REPROCESSINGPLANNED

  END_GROUP              = ECSDATAGRANULE

  GROUP                  = COLLECTIONDESCRIPTIONCLASS

    OBJECT                 = VERSIONID
      NUM_VAL              = 1
      VALUE                = 1
    END_OBJECT             = VERSIONID

    OBJECT                 = SHORTNAME
      NUM_VAL              = 1
      VALUE                = "MOD04L_O"
    END_OBJECT             = SHORTNAME

  END_GROUP              = COLLECTIONDESCRIPTIONCLASS

  GROUP                  = INPUTGRANULE

    OBJECT                 = INPUTPOINTER
      NUM_VAL              = 20
      VALUE                =
("MOD04_L2.A1997226.0725.002.1999204060629.hdf",
"MOD04_L2.A1997226.0730.002.1999204060716.hdf",
"MOD04_L2.A1997226.0735.002.1999204060620.hdf",
"MOD04_L2.A1997226.0740.002.1999204060637.hdf",
"MOD04_L2.A1997226.0745.002.1999204060647.hdf",
"MOD04_L2.A1997226.0750.002.1999204060644.hdf",
"MOD04_L2.A1997226.0800.002.1999204195105.hdf",
"MOD04_L2.A1997226.0805.002.1999204194908.hdf",
"EOSAM1_1997-08-14.eph", "EOSAM1_1997-08-14.att")
    END_OBJECT             = INPUTPOINTER
```

```
 END_GROUP            = INPUTGRANULE

 GROUP                = PGEVERSIONCLASS

  OBJECT               = PGEVERSION
   NUM_VAL             = 1
   VALUE               = "2.3.0"
  END_OBJECT           = PGEVERSION

 END_GROUP            = PGEVERSIONCLASS

 GROUP                = RANGEDATETIME

  OBJECT               = RANGEENDINGDATE
   NUM_VAL             = 1
   VALUE               = "1997-08-14"
  END_OBJECT           = RANGEENDINGDATE

  OBJECT               = RANGEENDINGTIME
   NUM_VAL             = 1
   VALUE               = "08:45:02"
  END_OBJECT           = RANGEENDINGTIME

  OBJECT               = RANGEBEGINNINGDATE
   NUM_VAL             = 1
   VALUE               = "1997-08-14"
  END_OBJECT           = RANGEBEGINNINGDATE

  OBJECT               = RANGEBEGINNINGTIME
   NUM_VAL             = 1
   VALUE               = "07:04:58"
  END_OBJECT           = RANGEBEGINNINGTIME

 END_GROUP            = RANGEDATETIME

 GROUP                = ORBITCALCULATEDSPATIALDOMAIN

  OBJECT               =
ORBITCALCULATEDSPATIALDOMAINCONTAINER
   CLASS               = "1"

   OBJECT               = EQUATORCROSSINGDATE
    CLASS               = "1"
    NUM_VAL             = 1
    VALUE               = "1997-08-14"
   END_OBJECT           = EQUATORCROSSINGDATE

   OBJECT               = EQUATORCROSSINGTIME
    CLASS               = "1"
    NUM_VAL             = 1
    VALUE               = "07:52:01.841571Z"
```

```
    END_OBJECT            = EQUATORCROSSINGTIME

    OBJECT                = ORBITNUMBER
      CLASS               = "1"
      NUM_VAL             = 1
      VALUE               = 24565
    END_OBJECT            = ORBITNUMBER

    OBJECT                = EQUATORCROSSINGLONGITUDE
      CLASS               = "1"
      NUM_VAL             = 1
      VALUE               = 0.700127
    END_OBJECT            = EQUATORCROSSINGLONGITUDE

  END_OBJECT            =
ORBITCALCULATEDSPATIALDOMAINCONTAINER

 END_GROUP              = ORBITCALCULATEDSPATIALDOMAIN

 GROUP                  = SPATIALDOMAINCONTAINER

  GROUP                  = HORIZONTALSPATIALDOMAINCONTAINER

    GROUP                  = GPOLYGON

      OBJECT                 = GPOLYGONCONTAINER
        CLASS                = "1"

        GROUP                  = GRINGPOINT
          CLASS                = "1"

          OBJECT                 = GRINGPOINTLONGITUDE
            NUM_VAL              = 6
            CLASS                = "1"
            VALUE                = (160.828918, 57.056137, 44.058830, -
1.859100, 34.022232, -114.758156)
          END_OBJECT             = GRINGPOINTLONGITUDE

          OBJECT                 = GRINGPOINTLATITUDE
            NUM_VAL              = 6
            CLASS                = "1"
            VALUE                = (69.526398, 23.441734, -65.489883, -
58.831890, 26.757355, 81.381241)
          END_OBJECT             = GRINGPOINTLATITUDE

          OBJECT                 = GRINGPOINTSEQUENCENO
            NUM_VAL              = 6
            CLASS                = "1"
            VALUE                = (1, 2, 3, 4, 5, 6)
          END_OBJECT             = GRINGPOINTSEQUENCENO

        END_GROUP              = GRINGPOINT
```

```
        GROUP                  = GRING
          CLASS                = "1"

          OBJECT               = EXCLUSIONGRINGFLAG
            NUM_VAL            = 1
            CLASS              = "1"
            VALUE              = "N"
          END_OBJECT           = EXCLUSIONGRINGFLAG

        END_GROUP              = GRING

      END_OBJECT               = GPOLYGONCONTAINER

    END_GROUP                  = GPOLYGON

  END_GROUP                    = HORIZONTALSPATIALDOMAINCONTAINER

 END_GROUP                     = SPATIALDOMAINCONTAINER

 GROUP                         = ASSOCIATEDPLATFORMINSTRUMENTSENSOR

  OBJECT                       =
ASSOCIATEDPLATFORMINSTRUMENTSENSORCONTAINER
    CLASS                      = "1"

    OBJECT                     = ASSOCIATEDSENSORSHORTNAME
      CLASS                    = "1"
      NUM_VAL                  = 1
      VALUE                    = "CCD"
    END_OBJECT                 = ASSOCIATEDSENSORSHORTNAME

    OBJECT                     = ASSOCIATEDPLATFORMSHORTNAME
      CLASS                    = "1"
      NUM_VAL                  = 1
      VALUE                    = "AM-1"
    END_OBJECT                 = ASSOCIATEDPLATFORMSHORTNAME

    OBJECT                     = ASSOCIATEDINSTRUMENTSHORTNAME
      CLASS                    = "1"
      NUM_VAL                  = 1
      VALUE                    = "MODIS"
    END_OBJECT                 = ASSOCIATEDINSTRUMENTSHORTNAME

  END_OBJECT                   =
ASSOCIATEDPLATFORMINSTRUMENTSENSORCONTAINER

 END_GROUP                     =
ASSOCIATEDPLATFORMINSTRUMENTSENSOR

END_GROUP                      = INVENTORYMETADATA
```

END

26.3.5 Mail Template

```
From ts2cm@t1ins01.vatc.ecs.nasa.gov Thu Jan  8 16:32 EST 1998
Received: from t1ins02 (t1ins02.vatc.ecs.nasa.gov [198.118.232.41])
          by t1ins01.vatc.ecs.nasa.gov (8.8.6/8.8.4) with SMTP
        id QAA10950 for <dps@t1ins01.vatc.ecs.nasa.gov>; Thu, 8 Jan 1998
16:32:01 -0500 (EST)
Received: by t1ins02 (SMI-8.6) id QAA19469; Thu, 8 Jan 1998 16:32:01 -0500
From: Code2 Install Team <ts2cm@t1ins01.vatc.ecs.nasa.gov>
Date: Thu, 8 Jan 1998 16:32:01 -0500
Message-Id: <199801082132.QAA19469@t1ins02>
Subject: ECS Notification
To: dps@t1ins01.vatc.ecs.nasa.gov
Content-Type: text
Content-Length: 79
Status: RO

UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[VTC:DSSDSRV]:20:SC:MODPTQKM.001:504
3
```

UR to be replaced with new UR.

26.3.6 Examining the Validity of Product Metadata

1 Login to the Science Data Server platform (p0acs03 in the PVC).

2 "**cd /usr/ecs/<*MODE*>/CUSTOM/logs**"

3 Open, in a text editor, EcDsScienceDataServer.ALOG

4 Search for the string "**Begin Metadata Validation (*ESDTShortName*)**".

Between this location in the file and the corresponding End Metadata Validation, you will find errors, if present, in the metadata test products. Also, there will be a message stating if metadata is Valid, metadata has Warnings or metadata is Invalid. This will depend on the nature of any errors, if present, in the product.

26.3.7 Insert Testing of Products

This same system is used to test products that are the DAP (Delivered Algorithm Package) products delivered to a DAAC for SSI&T. Typically, the products are a compressed TAR file with a file name of the form *string*.tar.Z After initial processing, the product is broken apart into its components, and those components will be subsequently processed and used based on their intended function.

The **insert** service is used to put the test .hdf file into the Data Server. Once the test .hdf file is in the Data Server, the **acquire** service is used to retrieve it.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The product ESDT has been installed on the Data Server. (See section 5.3 on how to install ESDTs.)
2. The SSIT Manager is running. (See section 6 on how to bring it up)

To Insert a DAP to the Science Data Server, execute the following steps:

1 From the SSIT Manager, click on the **Tools** menu, then choose **Data Server** and then **Insert Test Dynamic**.

- An xterm with title "SSIT: PGE Test Dynamic Input File Insertion" will be displayed.

2 At the program prompt **Configuration filename? (enter for default: ../..cfg/EcDpAtInsertTestFile.CFG)**, press **Return**.

3 At the program prompt **ECS Mode of operations?**

- Type in the **<mode>** you are working in. For example, **TS1** or **OPS.** Press **Return.**

4 At the program prompt **ESDT short name for the file to insert?** type ***ESDTShortName***, press **Return**

- The ***ESDTShortName*** is the ShortName for the product to test.

5 At the program prompt **ESDT Version for the file(s) to insert?** Type in the ESDT version and press **Return.**

6 At the program prompt **Is there more than one data file to this Dynamic Granule (Y = Yes, N = No)? (enter for default: N)?**

7 At the program prompt **Single Filename to Insert? (including FULL path)** type ***pathname***/faked.hdf, press

- The ***pathname/faked.hdf*** is the full path name and file name of a test .hdf file that is inserted into the Data Server. See description of this file below.

8 At the program prompt **Associated ASCII Metadata Filename to Insert (including FULL path)** , Type ***pathname/ESDT_ShortName.met*** and press **Return**.

- ***pathname*** is full name of the path and ***ESDT_ShortName.met*** is the name of the .met file of the product to be tested.

9 If the **insert** is successful, a **UR** is returned. Create a mail message with the name ESDT_ShortName.mail and copy the mail header described below. Copy and paste the **UR** into this .mail file and save it.

10 At the program prompt **Hit return to run again, 'q <return>' to quit:** type **q** and press **Return** to quit or just press **Return** to test **insert**s of additional products.

- If continuing, repeat steps 2 through 9.

26.4 Science Software Configuration Management

This section describes procedures for handling the configuration management of science software delivered to the DAACs for SSI&T. The COTS tool used for this purpose is ClearCase® by Atria Software, Inc. ClearCase can be run from the command line and via a graphical user interface (GUI).

The CM Administrator and System Administrator are key players in the SSI&T process. The CM Administrator receives the science software from the Science Data Specialist, places these files into a directory and request that the System Administrator place the files under configuration control by using the ClearCase tool. The science software is then tested by the SSI&T team and once the science software has successfully been tested, and upon direction from the CCB, the files are distributed to the Production Planner for placement on production server.

The CM and System Administrator need a good understanding of the ClearCase tool. ClearCase will be used to create a view, create a new directory, import files into the temporary subdirectories, and check-in and check-out files.

26.4.1 ClearCase Overview

All data managed under ClearCase are stored in Versioned Object Bases (VOBs), which are the "public" storage areas and Views, which are the "private storage areas. VOBs are data structures that can only be created by the CM administrator using the mkvob ("make vob") command. A VOB is mounted as a file system and when viewed through a view, it appears as a standard UNIX directory tree structure. This file system, accessed through its mount point, has a version-control dimension that contains file elements and versions of file elements. Once reviewed, the System Administrator will place these files under configuration control. In order to accomplish this task, a view must be created in ClearCase. A view is necessary in order to make visible and accessible files and directories that have been checked in to a VOB.

Data that are under configuration management in ClearCase are said to be "checked in". In order to alter a checked-in data element (e.g. a file) to make a newer version of it, the data element must first be "checked out". Once the change has been made to the checked- out version, it is checked in again. The VOB will then contain both versions of the data element and either can be retrieved at a later date.

In general, executable binary files, object files, and data files should not be checked into ClearCase. Binary and object files are not stored efficiently in ClearCase; data files for software may be extremely large and a VOB is typically not sized for this.

Files that should be checked into ClearCase include source code, scripts, makefiles, assorted build and run scripts, documentation and other ASCII files.

The administrator in charge of the VOB is referred to as the VOB administrator (VA).

All ClearCase procedures assume that the user's umask is set to 002.

For the Graphical User Interface version of the following procedures please refer to Operations Tools Manual for the EMD Project, 609-EMD-001.

A Versioned Object Base is defined by the following characteristics:

- A mountable file system that stores version-controlled data, such as source files, binary files, object libraries, WYSIWYG documents, spreadsheets and anything that can be stored in the UNIX file system.
- Can be mounted on some or all workstations
- Several VOBs may exist on a machine or on different machines on a network.
- When mounted as a file system of type MFS, a VOB can be accessed with standard UNIX and ClearCase tools.
- The ClearCase file system is transparent.
- Created by the CM administrator

A VOB is comprised of:

- Storage area for versioned files, derived objects and clear text files.
- Database (live, shadow and log file).

26.4.2 Creating a View in ClearCase

In order to make files and directories that are in a ClearCase VOB visible and accessible a ClearCase view must set. A ClearCase view need only be created once. Once created, the view can be set at the beginning of each user session. Multiple views for a single user may be created.

In order for the SSI&T tools under the SSIT Manager to have access to the ClearCase VOB, the ClearCase view must be set *before* the SSIT Manager is run.

A view is defined by the following characteristics:

- A working context for an individual developer or closely coordinated group.
- Can be used to access any VOB or multiple VOBs.
- Selects versions of VOB directories and files to display.
- Allows developer to work without interfering with other developers.
- Not a set of files but a way of seeing shared elements.
- Each user may have multiple views for new development, bug fixing or porting activities.

A view is comprised of:

- View storage area (typically in a local machine) - private storage for checked-out files, derived objects and private files.
- Configuration Specification - set of rules that determine the version of a file the view will see.
- View-tag - Name given to the view (ex. angies_view), view-tags are registered in /urs/adm/atria/view_tags.
- Objects stored in a view:

- Checked-out versions of file elements.
- Unshared derived objects.

The ClearCase procedures can either be run from the UNIX command line or from the File Browser Screen. The SSI&T Training will only cover the UNIX command line procedures. The corresponding GUI procedures are included in the Training Material for future reference.

The following procedure not only will create a view, but will also allow creation of a subdirectory where new science software files may be stored.

Assumptions:

1. ClearCase is available.
2. A Versioned Object Base (VOB) has been created.

26.4.2.1 Creating a View in ClearCase Using Command Lines

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

2 Enter the **password** then press the **Enter** key.

3 At a UNIX prompt type **cleartool lsview**, then press the **Enter** key.

- The **lsview** command displays the pathname to the storage location of the views.

4 At a UNIX prompt type **cleartool mkview -tag *ViewName ViewPath/ViewName*.vws,** then press the **Enter** key.

- The ***ViewPath*** is the full path to the directory where views are stored. The SA should supply this information. A typical example is **/net/mssg1sungsfc/viewstore/.**
- The *ViewName* is the user selected name for the view. The file name for the view must end in ".vws".

For future reference, the corresponding ClearCase GUI procedures are included in the following section.

26.4.2.2 Creating a View in ClearCase using the File Browser Screen

Selecting a view listed in the View Tag Browser screen brings up the File Browser, or main screen.

- Displays the directory name of the current VOB, just below the toolbar.
- Displays the content of the directory in the space below the directory's name.

Procedures

1 The user should log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

- Cursor moves to the **Password** field.

2 Type the **password** then press the **Enter** key.

3 Invoke ClearCase by typing **xclearcase &** on the UNIX command line then press the **Enter** key.

- The ClearCase **Transcript** screen is displayed as the View Tag Browser loads.
- The ClearCase **View Tag Browser** screen is displayed listing available views.

4 To create a view for checking in the software change package, select a known View and press the **Enter** key.

- The File Browser window is displayed.

5 Select **File→Execute→Single Command.**

- The String Browser window is displayed.
- The prompt Enter shell command to run is displayed.

6 Invoke the make view command by typing **mkview [filename]** on the UNIX command line and press the **Enter** key.

- The **tempdisp** window appears.
- The **View [filename] Created Successfully** and the **Cache Updated for View [filename]** prompts are displayed.

7 Close the **tempdisp** window by clicking on the window and press the **Enter** key.

- The **tempdisp** window closes.

8 Select **View →List** from the menu.

- The **View Tag Browser** is displayed.

9 Find the new view by scrolling through the list until the new view is observed.

26.4.3 Setting a View in ClearCase

In order to make files and directories that are in a ClearCase VOB visible and accessible, a ClearCase view must be set. Only one view can be set (active) at a time.

26.4.3.1 Setting a View in ClearCase Using Command Lines

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

2 At a UNIX prompt type **cleartool setview *ViewName*** where ***ViewName*** is the user's view created in the previous section, then press the **Enter** key.

26.4.3.2 Setting a View Using the File Browser Screen in ClearCase

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

- Cursor moves to the **Password** field.

2 Type the **password** then press the **Enter** key.

3 Invoke ClearCase by typing **xclearcase &** on the UNIX command line then press the **Enter** key.

- The ClearCase **Transcript** screen is displayed as the View Tag Browser loads.
- The ClearCase **View Tag Browser** screen is displayed listing available views.

4 To set a view, select a known View and press the **Enter** key.

- The File Browser window is displayed.

5 Select **File→Execute→Single Command.**

- The String Browser window is displayed.
- The prompt **Enter** shell command to run is displayed.

6 Invoke the set view command by typing **setview ViewName** on the UNIX command line and press the **Enter** key.

- **ViewName** is the name of the view to set.

26.4.4 Creating a New Directory

In cases where a new directory needs to be created and placed in ClearCase, the user will activate ClearCase and create a new directory. This type of procedure is necessary only if a new directory is required.

The following is a list of tools and/or assumptions:

1. A VOB has been created at the UNIX directory.
2. A view has been created.

26.4.4.1 Creating a New Directory in ClearCase Using Command Lines

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

2 Enter the **password** then press the **Enter** key.

3 At a UNIX prompt type **cleartool setview *ViewName***, then press the **Enter** key.

- The ***ViewName*** is the user's view.

4 At a UNIX prompt type **cleartool lsvob,** then press the **Enter** key.

- This command lists all the VOBs and allows the identification of the SSI&T VOB.

5 At a UNIX prompt type **cd *pathname***, then press the **Enter** key.

- The ***pathname*** is the full path name of the parent directory in the VOB in which the new directory is to be added. . For example, if a new directory is to be added under /VOB1/pge4, type **cd /VOB1/pge4**, . (note the space and then "dot" at the end of the command).

6 At a UNIX prompt type **cleartool checkout -nc .** then press the **Enter** key.

- This command checks out the current directory. Note the dot for the directory.
- The **-nc** is a keyword used when no comments are to be made for this action.

7 At a UNIX prompt type **cleartool mkdir -nc *dirname*,** then press the **Enter** key.

- The ***dirname*** is the name of the new directory being created.

8 At a UNIX prompt type **cleartool checkin -nc *dirname*,** then press the **Enter** key.

- This command checks in the new directory named ***dirname***.

9 At a UNIX prompt type **cleartool checkin -nc .** then press the **Enter** key.

- This command checks in the current directory.

26.4.4.2 Entering a New Directory into ClearCase Using the File Screen Browser

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

- Cursor moves to the **Password** field.

2 Type the **password** then press the **Enter** key.

3 Invoke ClearCase by typing **xclearcase &** on the UNIX command line then press the **Enter** key.

- The ClearCase **Transcript** screen is displayed as the View Tag Browser loads.
- The ClearCase **View Tag Browser** screen is displayed listing available views.

4 Select **File→Execute→Single Command.**

- The String Browser window is displayed.
- The prompt **Enter shell command to run** is displayed.

5 Invoke the make directory element by typing **mkdir [filename]** on the UNIX command line and press the **Enter** key.

6 Invoke the make element command by typing **mkelem [directory name]** on the UNIX command line and press the **Enter** key.

7 Type into the directory input box of the **File Browser** the name of the directory in the VOB to be checked out, press the **Enter** key, then follow the menu path **Version→Checkout→Reserved: no comment**.

- In order to add new files to ClearCase, the directory in which the files are to be added must be checked out first.
- ClearCase forces the checkout onto a maintenance branch to isolate the maintenance activity.
- If someone else has already checked out the directory, permission to check out the directory is denied. A separate shell window is displayed.

8 Cancel the checkout of the element if it is decided that no changes are to be made by typing into the directory input box of the **File Browser** the name of the directory to be checked in, press the **Enter** key, then follow the menu path **Version→Uncheckout→Unreserved: no comment**,

9 On the **File Browser** screen, follow the menu path **File→Exit**.

- The ClearCase Graphical User Interface session is closed.

26.4.5 Importing Files into ClearCase

Once the user has created a directory to place the science software files, ClearCase can be used to place a single file or multiple files in a UNIX directory structure under CM.

The following is a list of tools and/or assumptions:

1. A VOB and subdirectory are created to hold these files.
2. No object files or executables exist in the source code directory.
3. The PGE was received with a directory structure that contains various types of files.
4. These files will be entered into ClearCase and will maintain the same directory structure as the delivery structure.

26.4.5.1 Importing a Single File into ClearCase

Procedure:

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

- Cursor moves to the **Password** field.

2 Type the **password** then press the **Enter** key.

3 At a UNIX prompt, type **cleartool setview *ViewName***, press **Enter**

- The ***ViewName*** is the name of the ClearCase View.

4 At the UNIX prompt, type **cd *pathname***, then press the **Enter** key.

- The ***pathname*** is the full path name of the subdirectory in the VOB into which the file is to be checked in. . For example, to check a file into the VOB directory /VOB1/pge2/scripts/, type **cd /VOB1/pge2/scripts/** .
- If the desired directory cannot be seen, it could mean that the view has not been set or the properties of the view do not allow the directory to be seen; check with the CM Administrator.

5 At a UNIX prompt, type **cp *pathname/filename* .**, press **Enter** (note the space and then "dot" at the end of the command)**.**

- The ***pathname*** is the full path name to the directory where the file to be checked in exists and ***filename*** is the file name of the file to be checked in.
- This command copies a file over into the VOB area in preparation for checking it in. . For example, to copy over a file named MISR_calib.c in directory **/pge/pge34/** to be checked in, type **cp pge/pge34/MISR_calib.c .**,

6 At the UNIX prompt, type **cleartool checkout -nc .**, press **Enter** (note the space and then "dot" at the end of the command).

- This command checks out the current directory (represented by the "dot") from ClearCase.
- Adding a new file (or element) to a directory represents a modification of the directory. Hence, the directory must be checked out before a file can be checked in.

7 At a UNIX prompt, type **cleartool mkelem -nc *filename***, then press the **Enter** key.

- The ***filename*** is the name of the file that was copied over in step 5 and is the file that will be checked into ClearCase.
- This command creates a ClearCase element from the file in preparation for checking it in.
- The **-nc** flag means “no comment”; it suppresses ClearCase from prompting for a comment to be associated with the make element step.

8 At the UNIX prompt, type **cleartool checkin -nc *filename***, then press the **Enter** key.

- The ***filename*** is the name of the file to be checked into ClearCase.
- This command performs the check in of the file.
- The **-nc** flag means “**no comment**”; it suppresses ClearCase from prompting for a comment to be associated with the checkin step. To put comment in, use -c "comment text". For example, **cleartool checkin -c "adding version 2" filename**. By default, **cleartool** expects a comment and in the case where neither option is used, **cleartool** will prompt for a comment. In such case, simply add the comment text and a dot at the end (indicating the end of comment).]

9 At the UNIX prompt, type **cleartool checkin -nc .**, press **Enter** (note the space and then “dot” at the end of the command).

- This command checks in the current directory (represented by the “dot”) into ClearCase.
- The adding of an element (here, a file) represents a modification to the directory and hence, the new version of the directory must be checked back in.
- The **-nc** flag means “no comment”; it suppresses ClearCase from prompting for a comment to be associated with the checkin step.

26.4.5.2 Importing Multiple Files into ClearCase

The DAP for the synthetic PGE contains only one source code module and a minimal number of other files. A real PGE will generally contain many source files, header files, and multiple other types of files stored in a standard type of directory structure that is retained when the PGE is packed into the tar file. The script provided by ClearCase is used for the purpose of making another load script to enter all of the DAP files along with the directory structure at one time. The final step of running the load script can only be performed by the DAAC Administrator.

The following procedure explains how to place the entire contents of a UNIX directory structure under ClearCase. A UNIX directory structure refers to all the files and subdirectories under some top-level directory.

This procedure is geared toward science software deliveries. In such cases, science software is delivered in the form of a UNIX *tar* files. A *tar* file has been unpacked (un*tar*-red) and the contents are to be placed under ClearCase configuration management.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

The following is a list of tools and/or assumptions:

1. A VOB and subdirectory are created to hold these files.
2. A ClearCase view is **not** required to perform this procedure.

Importing Multiple Files Into ClearCase

1 At a UNIX prompt, type **cd *ParentPathname***, then press the **Enter** key

- The ***ParentPathname*** is the path name of the directory that *contains* the directory structure to be brought into ClearCase. This is *not* the VOB.

2 At the UNIX prompt, type **clearcvt_unix -r *DirName***, then press the **Enter** key.

- The *DirName* is the name of the directory in which it and everything below it is to be brought into ClearCase.
- A conversion script will be then be created. The -r causes all subdirectories to be recursively included in the script created.

3 Contact the VOB Administrator and request that the utility script cvt_script be run on the script created in step 2.

- The VOB Administrator is the only one who can run the cvt_script because it modifies the VOB.

4 At this time the user logs out from this workstation. The VOB Administrator completes the procedure.

- The remaining steps are accomplished by the VOB Administrator.

5 The VOB Administrator logs into the AIT Sun workstation by typing **username** then press the **Enter** key.

- Cursor moves to the **Password** field.

6 Type the **password** then press the **Enter** key.

7 Invoke ClearCase by typing **xclearcase &** on the UNIX command line then press the **Enter** key.

- The ClearCase **Transcript** screen is displayed as the View Tag Browser loads.
- The ClearCase **View Tag Browser** screen is displayed listing available views.

8 To create a view for checking in the software change package, select a known View and press the **Enter** key. If you are using an existing view, select the desired existing view and proceed to step 14.

- The File Browser window is displayed.

9 Select **File→Execute→Single Command.**

- The String Browser window is displayed.
- The prompt Enter shell command to run is displayed.

10 Invoke the make view command by typing **mkview [filename]** on the UNIX command line and press the **Enter** key.

- The **tempdisp** window appears.
- The **View [filename] Created Successfully** and the **Cache Updated for View [filename]** prompts are displayed.

11 Close the **tempdisp** window by clicking on the window and press the **Enter** key.

- The **tempdisp** window closes.

12 Select the VOB where the software change package is to be imported then press the **Enter** key.

13 To create a subdirectory for the software change package in that VOB, which is a modification to the parent directory (for the VOB) the parent directory must be checked out by following the menu path **Version→Checkout→Reserved: no comment**.

- In order to add new files to ClearCase, the directory in which the files are to be added must be checked out first.
- ClearCase forces the checkout onto a maintenance branch to isolate the maintenance activity.
- If someone else has already checked out the directory, permission to check out the directory is denied.
- A separate shell window is displayed.

14 Start a shell process in a separate window by clicking on the shell icon button of the **File Browser** toolbar.

- A separate shell window is displayed.

15 To run the script, type **cvt_script** then press the **Enter** key.

- The VOB Administrator is the only person who can run the **cvt_script** because it modifies the VOB.

16 To check in the new directory, type into the directory input box of the **File Browser** screen: **path** [where **path** is the full path identification for the new directory (**directoryname**)], then press the **Enter** key. Then select **Versions→Checkin** from the menu.

17 To check in the parent directory (for the VOB), type into the directory input box of the **File Browser** screen: **VOBpath** (where **VOBpath** is the full path identification for the parent directory), then press the **Enter** key. Then select **Versions→Checkin** from the menu.

18 On the **File Browser** screen, follow menu path **File→Exit**.

- The ClearCase Graphical User Interface session is closed.

26.4.6 Checking Out a File from ClearCase

If a configured file requires modification, then the file needs to be checked out of the configured directory and placed in a user directory. This will allow the file(s) to be modified.

The following is a list of tools and/or assumptions:

1. The file or directory must be an element created in ClearCase.
2. The view should be configured to ensure the correct version of the file or directory is seen.

26.4.6.1 Checking Out an Element/File from the Command Line

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

2 Enter the **password** then press the **Enter** key.

3 At a UNIX prompt type **cleartool setview *ViewName***, then press the **Enter** key.

- The ***ViewName*** is the name if the user's view.

4 At a UNIX prompt type **cleartool checkout -nc *element*** then press the **Enter** key.

- The ***element*** is the name of the file or directory that is to be checked out.
- The **-nc** flag means "no comment" which will suppress the ClearCase prompting for a comment to be associated with the check out step.

26.4.6.2 Checking Out an Element/File from the File Screen Browser

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

- Cursor moves to the **Password** field.

2 Type the **password** then press the **Enter** key.

3 Invoke ClearCase GUI by typing **xclearcase &** on the UNIX command line then press the **Enter** key.

- The ClearCase **Transcript** screen is displayed as the View Tag Browser loads.
- The ClearCase **View Tag Browser** screen is displayed listing available views.

4 To check out the directory where the controlled files were place, type into the directory input box of the **File Browser** screen: **path** [where **path** is the full path identification for the directory (**directoryname**)], then press the **Enter** key. Then select **Versions**→**Checkout** from the menu.

5 Select **File**→**Execute**→**Single Command.**

- The String Browser window is displayed.
- The prompt **Enter shell command to run** is displayed.

6 To determine editing privileges, type **ls -l,** then press the **Enter** key.

- A prompt displaying read/write/execute privileges will be displayed. There will be three groupings:
- **User Group Others**
- **r=read, w=write, x=execute**

7 If you have editing/execute privileges, you can revise the contents of the file with any text editor.

8 To check in a controlled file, select **Versions**→**Checkin** from the menu.

- The file/directory will be checked in to ClearCase and the version will be updated.

26.4.7 Checking a Modified Element into ClearCase

This procedure explains how to check in a modified element to ClearCase. An element refers to a directory or file in ClearCase, that is, under configuration management. Modifications made to a file or directory cannot be saved in ClearCase unless the file or directory had been checked out first.

The following is a list of tools and/or assumptions:

1. A VOB exits and is mounted at a known UNIX directory.
2. A ClearCase view exists for the SSI&T operator.
3. The element or file has been checked out and modified.

4. The modified file is now in the user's directory on the VOB from which it was checked out.

26.4.7.1 Checking a Modified Element/File into ClearCase

1 Log into one of the AIT Sun workstations by typing: **username** then press the **Enter** key.

- Cursor moves to the **Password** field.

2 Type the **password** then press the **Enter** key.

3 At a UNIX prompt, type **cleartool setview *ViewName***, then press the **Enter** key.

- The ***ViewName*** is the name of the user's view.

4 At the UNIX prompt, type **cleartool checkin -nc *filename***, then press the **Enter** key.

- The ***filename*** is the name of the file (full path name allowed) that is to be checked out (and later modified).
- The **-nc** flag means "no comment"; it suppresses ClearCase from prompting for a comment to be associated with the check out step.
- This command checks in the current directory.

5 This step is optional; it is performed when ClearCase does not accept a checkin because the element was not modified. In this case, the check out must be canceled. At a UNIX prompt, type **cleartool uncheckout -nc *filename***, then press the **Enter** key.

- The ***filename*** is the name of the file or directory (full path name allowed) checked out.
- This command cancels the check out of an element/file.

26.5 Standards Checking of Science Software

The purpose of standards checking is to verify that the source files of the science software are compliant with the ESDIS Data Production Software Computing Facility (SCF) Standards and Guidelines document.

26.5.1 Checking FORTRAN 77 ESDIS Standards Compliance

The ESDIS Data Production Software Computing Facility (SCF) Standards and Guidelines document requires all FORTRAN 77 code to be compliant with the ANSI FORTRAN 77. The COTS used for this task is FORCHECK.

The following is a list of tools and/or assumptions:

Assumptions:

1. The FORTRAN 77 science software source code is available, accessible, and has read permissions for the user.

2. SSIT Manager is available for use.

3. FORCHECK is available only on the AIT Suns.

To check for ESDIS standards compliance in FORTRAN 77 code, execute the procedure steps that follow:

1 If not already on an AIT Sun, log into one from your machine.
- Once logged onto proper Sun, remember to set the DISPLAY environmental variable to point to your X Window screen.

2 If required, at the UNIX prompt on the AIT Sun, type **cleartool setview *ViewName***, press **Return**.
- The ***ViewName*** is the name of a view allowing the FORTRAN 77 source files to be accessible.
- This step is only necessary if any of the FORTRAN 77 source files are in ClearCase (in the VOB under configuration management).

3 If your general environment setup does not include transparent access to the SSIT Manager GUI, then you need to set that up. One way to do it is as follows:
- Set up an alias, manually or from shell script, to set up preliminary environment. At UNIX prompt, type **alias do_buildrc "source /usr/ecs/TS1/CUSTOM/bin/DPS/.buildrc"**
- Set up an alias, manually or through shell script, to invoke SSIT Manager. At UNIX prompt, type **alias do_ssit_man "/usr/ecs/TS1/CUSTOM/bin/DPS/EcDpAtMgr ConfigFile /usr/ecs/TS1/CUSTOM/cfg/EcDpAtMG.CFG ecs_mode TS1& "**

4 Set up the preliminary environment (do_buildrc). This only needs to be done once per session. Then, run SSIT Manager (do_ssit_man).
- Type **do_buildrc**
- Type **do_ssit_man**

5 Once the SSIT Manager comes up, the following steps need to be taken to invoke FORCHECK
- From the top menu bar, select **Tools.**
- From the Tools menu, select **Standards Checkers.**
- From the Standards Checkers menu, select **FORCHECK.**

6 A separate FORCHECK window will now open.
- The user will be prompted for input. The first prompt will be *global option(s) and list file?*

- The second prompt will be
  local option(s) and file(s)?
- The second prompt will be repeated until there is a blank line and carriage return.
- In order to understand what the proper responses should be, the user is encouraged to find hardcopy documentation for FORCHECK or to use the UNIX man facility and type *man forchk* .

7 At the UNIX prompt on the AIT Sun, type **vi *FORCHECKoutput***, press **Return**.

- The ***FORCHECKoutput*** is the file name for the output file produced in step 6.
- The ***FORCHECKoutput*** file will contain any warnings, errors, and other messages from FORCHECK. A summary will be at the bottom of the file.
- Any text editor may be used for this procedure step.

8 At the UNIX prompt on the AIT Sun, type **vi *ListFile***, press **Return**.

- The ***ListFile*** is the file name for the list file specified at the FORCHECK prompt.
- The ***ListFile*** file will contain FORCHECK messages similar to the ***FORCHECKoutput*** file embedded in the source code listing.
- Any text editor may be used for this procedure step.

26.5.2 Checking for ESDIS Standards Compliance in Fortran 90

This procedure describes how to use the Fortran 90 compiler flags on the SPR SGI machines to check science software written in Fortran 90 for ESDIS standards compliance.

Unlike with FORTRAN 77, no COTS tool is used to check Fortran 90 science software. Instead, this procedure describes how to use the compiler to perform the checking (ESDIS standards for Fortran 90 are ANSI). Since the Fortran 90 compiler is used, the checking for standards compliance can be naturally tied in with building the science software (since this procedure will produce object files suitable for linking). However, in this procedure, the building of the software (compiling *and* linking) is deferred to a later procedure.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The Fortran 90 science software source code is available, accessible, and has read permissions for the user.
2. Required Status Message Facility (SMF) files have been compiled.
3. The C shell (or a derivative) is the current command shell.
4. The Fortran 90 compiler is available on the SPR SGI.

To check for ESDIS standards compliance in Fortran 90 code, execute the procedure steps that follow:

1 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to the SPR SGI.

- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then telnet to the SPR SGI.

2 At the UNIX prompt on the SPR SGI, set up the proper environment for the compiler to be used by typing **source *ToolkitPathname* /bin/*sgiXX*/pgs-dev-env.csh .**

- ***ToolkitPathname*** is the home directory of the desired SDP Toolkit version .
- The directory ***sgiXX*** should be replaced with **sgi32** or **sgi64** as appropriate for the specific compiler desired.
- For example, on the PVC platform p0spg01, type **source /data3/ecs/TS1/CUSTOM/daac_toolkit_f90/TOOLKIT/bin/sgi64/pgs-dev-env.csh .** This will set up the various environment parameters, such as PGSHOME, to enable the 64 bit version of the FORTRAN 90 compiler to be run.

3 If required, at the UNIX prompt on the SPR SGI, type **cleartool setview *ViewName***, press **Return**.

- The ***ViewName*** is the name of a view allowing the Fortran 90 source files to be accessible.
- This step is only necessary if any of the Fortran 90 source files are in ClearCase (in the VOB under configuration management).

4 At the UNIX prompt on the SPR SGI, type **cd *SrcPathname***, press **Return**.

- The ***SrcPathname*** is the full path name to the location of the Fortran 90 source files to be checked.
- The ***SrcPathname*** will be in the ClearCase VOB is the Fortran 90 source files are checked into ClearCase.

5 At the UNIX prompt on the SPR SGI, type **f90 -c -ansi *[*-I$PGSINC*]* *[*-I$HDFINC*]* *[[*-IOtherIncFiles*]…]* *SourceFiles* >& *ReportFile***, press **Return**.

- The terms in square brackets (*[ ]*) are used to optionally specify locations of include and module (.mod) files. The **$PGSINC** already contains the SDP Toolkit include directory and **$HDFINC** already contains the HDF include directory. The **OtherIncFiles** represents one or more additional include or module directories.
- The ***SourceFiles*** is a list (space delimited) of Fortran 90 source files or a wildcard template (*e.g.* *.f90).
- The **>&** is a C shell construct that causes standard error (where the output from the Fortran 90 compiler normally emerges) to be redirected to a file.
- The ***ReportFile*** is the file name under which to save the results of the compile process.
- The **-c** flag causes only compilation (no linking).

- The **-ansi** flag enables ANSI checking.
- Apply the terms in square brackets only as necessary. Do not include the brackets in the actual command. See example below.
- Do not use the **-I** option for include or module files that are in the standard directories or in the current directory.
- The makefile for the science software may contain the names of additional include files needed by the software.
- For example, type **f90 -c -I$PGSINC -I$HDFINC -I/ecs/modis/pge5/include/ *.f90 >& pge10.report**, press **Return**.

6 At the UNIX prompt on the SPR SGI, type **vi *ReportFile***, press **Return**.
- The ***ReportFile*** is the file name for the compilation results as produced in step 5.
- Any text editor may be used for this procedure step.

26.5.3 Checking for ESDIS Standards Compliance in C and C++

This procedure describes how to use the C compiler flags on the SPR SGI machines to check science software written in C for ESDIS standards compliance.

Unlike with FORTRAN 77, no COTS tool is used to check C science software. Instead, this procedure describes how to use the compiler to perform the checking (ESDIS standards for C are essentially ANSI). Since the C compiler is used, the checking for standards compliance can be naturally tied in with building the science software (since this procedure will produce object files suitable for linking). However, in this procedure, the building of the software (compiling *and* linking) is deferred to a later procedure.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The C science software source code is available, accessible, and has read permissions for the user.
2. Required Status Message Facility (SMF) files have been compiled.
3. The C shell (or a derivative) is the current command shell.
4. The C compiler is available on the SPR SGI.

To check for ESDIS standards compliance in C code, execute the procedure steps that follow:

1 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to the SPR SGI.
- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then telnet to the SPR SGI.

2 At the UNIX prompt on the SPR SGI, type **setenv PGSHOME *ToolkitPathname***, press **Return**. Then type, source **$PGSHOME/bin/*sgiX*/pgs-dev-env.csh**, press **Return**.

- The ***ToolkitPathname*** is the home directory of the desired SDP Toolkit version.
- The ***sgiX*** refers to the appropriate processor. For example, type **source $PGSHOME/bin/sgi/pgs-dev-env.csh**, press **Return**.

3 If required, at the UNIX prompt on the SPR SGI, type **cleartool setview *ViewName***, press **Return**.

- The ***ViewName*** is the name of a view allowing the C source files to be accessible.
- This step is only necessary if any of the C source files are in ClearCase (in the VOB under configuration management).

4 At the UNIX prompt on the SPR SGI, type **cd *SrcPathname***, press **Return**.

- The ***SrcPathname*** is the full path name to the location of the C source files to be checked.
- The ***SrcPathname*** will be in the ClearCase VOB is the C source files are checked into ClearCase.

5 At the UNIX prompt on the SPR SGI, type **cc -c -ansi *[*-I$PGSINC*] [*-I$HDFINC*] [[*-IOtherIncFiles*]…] SourceFiles* >& *ReportFile***, press **Return**.

- The terms in square brackets (*[ ]*) are used to optionally specify locations of include and module (.mod) files. The **$PGSINC** already contains the SDP Toolkit include directory and **$HDFINC** already contains the HDF include directory. The **OtherIncFiles** represents one or more additional include directories.
- The ***SourceFiles*** is a list (space delimited) of C source files or a wildcard template **(*e.g.* *.c or *.cpp).**
- The **>&** is a C shell construct that causes standard error (where the output from the C compiler normally emerges) to be redirected to a file.
- The ***ReportFile*** is the file name under which to save the results of the compile process.
- The **-c** flag causes only compilation (no linking).
- The **-ansi** flag enables ANSI checking.
- Apply the terms in square brackets only as necessary. Do not include the brackets in the actual command. See example below.
- Do not use the **-I** option for include files that are in the standard directories (*e.g.* /usr/include) or in the current directory.
- The makefile for the science software may contain the names of additional include files needed by the software.
- For example, type **cc -c –ansi-I$PGSINC -I$HDFINC -I/ecs/modis/pge5/include/ *.c >& pge10.report**, press **Return**.

6 At the UNIX prompt on the SPR SGI, type **vi *ReportFile***, press **Return**.

- The ***ReportFile*** is the file name for the compilation results as produced in step 5.
- Any text editor may be used for this procedure step.

26.5.4 Prohibited Function Checker

The use of certain functions in the PGE is prohibited. The Prohibited Function Checker is used to check C, FORTRAN 77 and FORTRAN 90 language source files for the occurrence of functions that are prohibited in the ECS DAAC production environment.

26.5.4.1 Checking for Prohibited Functions: Command-Line Version

This procedure describes using the command-line version of the Prohibited Function Checker to check science software for the prohibited functions.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The source files to be checked are available, accessible, and have read permissions for the operator.
2. Source files to be checked are C, FORTRAN 77, Fortran 90, C shell, Korn shell, Bourne shell, or Perl and have recognizable file name extensions.

To check for prohibited functions in delivered source files, execute the procedure steps that follow:

1 If required, at the UNIX prompt on an AIT Sun, type **cleartool setview *ViewName***, press **Return**.
- The ***ViewName*** is the name of a view allowing the source files to be accessible.
- This step is only necessary if any of the source files are in ClearCase (in the VOB under configuration management).

2 At the UNIX prompt on the AIT Sun, type **cd *SrcPathname***, press **Return**.
- The ***SrcPathname*** is the full path name to the location of the source files to be checked.
- The ***SrcPathname*** will be in the ClearCase VOB if the source files are checked into ClearCase.
- The ***SrcPathname*** can contain other directories that contain source files and/or more directories. The Prohibited Function Checker will search out all source files in subdirectories recursively.

3 At the UNIX prompt on the AIT Sun, type **/data3/ecs/TS1/CUSTOM/bin/DPS/EcDpAtMgrBadFunc ConfigFile /data3/ecs/TS1/CUSTOM/cfg/EcDpAtBA.CFG *FilesOrDirectories* > *ResultsFile***, press **Return**.

- The ***FilesOrDirectories*** is a list of source file names or directory names of directories containing source files.
- The ***ResultsFile*** is the file name for the results that are output.
- For example, type **/data3/ecs/TS1/CUSTOM/bin/DPS/EcDpAtMgrBadFunc ConfigFile /data3/ecs/TS1/CUSTOM/cfg/EcDpAtBA.CFG main.c utils/ > myOutput**, press **Return**. Here, **main.c** is a source file and **utils/** is a directory that contains other source files.

4 At the UNIX prompt on the AIT Sun, type **vi *ResultsFile***, press **Return**.

- The ***ResultsFile*** is the file name for the output results as produced in step 3.
- Any text editor may be used for this procedure step.

26.5.4.2 Prohibited Function Checker GUI Version

1 From the SSIT Manager, select **Tools → Standards Checkers → Prohibited Function Checker** from the menu.

- The Prohibited Function Checker GUI will be displayed.

2 In the Prohibited Function Checker GUI, click on the **Analyze** button.

- The File Selector GUI will be displayed.

3 Within the **Directories** subwindow, double click on the desired directory.

- Repeat this step until the directory with the source files to be checked are displayed in the **Files** subwindow.

4 Within the **Files** subwindow, click on the source files to be checked. Each file clicked on will be highlighted.

- To choose groups of contiguous files, hold down the left mouse button and drag the mouse.
- To choose non-contiguous files, hold down the Control key while clicking on file names.

5 In the File Selector GUI, click on the **OK** button.

- The File Selector GUI will disappear.
- The files selected in step 5 will be displayed in the Prohibited Function Checker GUI window as they are being checked.

6 In the Prohibited Function Checker GUI, click on the **Report** button.

- The **Report** GUI will be displayed.

- For each file, a list of prohibited functions found will be displayed.

7 Optionally, click on the **Print** button or the **Save** button.

- Choose **Save** to save the results to a file; choose **Print** to have the results printed on the default printer.
- Choosing **Save** will bring up a GUI labeled **Save To File**. Specify the directory and file name in which to save the results file.

8 Optionally, in the Prohibited Function Checker GUI, highlight one of the source files listed. Then click on **View**.

- The **Source Code** GUI will be displayed.
- Occurrences of prohibited functions found in that source file will be highlighted.
- Click on the **Next** button to bring into the window successive occurrences of prohibited functions (the **Next** button does not bring in the next source file).
- Click on the **Done** button to close the **Source Code** GUI. Other source files may be examined similarly, one at a time.

9 In the Prohibited Function Checker GUI, click on the **Quit** button.

- The Prohibited Function Checker GUI will disappear.
- This ends the session.

26.5.5 Checking for Prohibited Functions: GUI Version

This procedure describes using the GUI version of the Prohibited Function Checker to check science software for prohibited functions.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The SSIT Manager is running and that the source files to be checked are available, accessible, and have read permissions for the operator.
2. Source files to be checked are C, C++, Fortran 77, Fortran 90, C shell, Korn shell, Bourne shell, or Perl, and have recognized file name extensions (Table 26.5.5-1).

Table 26.5.5-1. File Name Extensions Recognized (1 of 2)

Language	File Name Extensions
C	.c, .h
C++	.cpp, .h
Fortran 77	.f, .f77, .ftn
Fortran 90	.f90
C Shell	.csh

Table 26.5.5-1. File Name Extensions Recognized (2 of 2)

Language	File Name Extensions
Korn Shell	.ksh
Bourne Shell	.sh
Perl	.pl

To check Prohibited Functions, execute the procedure steps that follow:

1. From the SSIT Manager, click **Tools-> Standards Checkers -> Prohibited Function Checker.** The Prohibited Function Checker GUI will be displayed.

26.5.6 Checking Process Control Files

The next task to accomplish is to check that the PCFs are syntactically correct and contain all necessary information for PGEs to run within the ECS DAAC production environment. Only one PCF can be associated with a PGE. The following procedure describes how to check PCFs for valid syntax and format, both using the GUI and the command line interface.

26.5.6.1 Checking Process Control Files GUI

The following is a list of tools and/or assumptions:

1. The SSIT Manager is running.
2. The Process Control File(s) are available, accessible, and have read permissions.

If the source code files to be checked are in a VOB in ClearCase, a view has been set before the SSIT Manager was started.

Checking Process Control Files GUI

1 From the SSIT Manager, select Tools → Standards Checkers → Process Control File Checker from the menu.

- The Process Control File Checker GUI will be displayed.

2 In the **Directories** subwindow, double click on the desired directory.

- Repeat this step until the directory with the PCF(s) to be checked is displayed in the Files window.
- Use the **Filter** subwindow to limit which files are displayed.

3 Within the **Files** subwindow, click on the PCF to be checked**.**

- The file clicked on will be highlighted.
- Only one PCF can be checked at a time.

4 Click on the **Check PCF** button.

- A GUI labeled **PCF Checker Results** will be displayed.
- Results will be displayed in this window.

5 Optionally, click on the **Save** button or on the **Print** button.

- Choose **Save** to save the results to a file; choose **Print** to have the results printed on the default printer.
- Choosing **Save** will bring up a GUI labeled **Save To File**. Specify the directory and file name in which to save the results file.
- Choosing **Print** and then clicking on the **OK** button will send the results to the default printer.

6 Click on the **Check Another** button or on the **Quit** button.

- Choosing **Check Another** allows another PCF to be checked. Repeat steps 2 through 5.
- Choosing **Quit** causes the Process Control File Checker GUI to disappear and ends the session.

26.5.6.2 Checking Process Control Files: Command-Line Version

This procedure describes using the command-line version of the Process Control File Checker to check process control files delivered with the science software.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The PCF files to be checked are available, accessible, and have read permissions for the operator.
2. You will need the command **pccheck.sh**. One way to see if this is available is to type **which pccheck.sh**, press **Return**. If a path is displayed, then the directory is in your path. On the PVC Sun platform **p0ais01,** the pathname for the command is **/ecs/formal/TOOLKIT/bin/sun5/pccheck.sh**. In this case, you will have to set a ClearCase view to access that area.

To check Process Control Files, execute the procedure steps that follow:

1 If required, at the UNIX prompt on an AIT Sun, type **cleartool setview *ViewName***, press **Return**.

- The ***ViewName*** is the name of a view allowing the Process Control File(s) to be accessible.
- This step is only necessary if any of the Process Control Files are in ClearCase (in the VOB under configuration management).

2 At the UNIX prompt on AIT Sun, type **cd *PCFpathname***, press **Return**.

- The ***PCFpathname*** is the full path name to the location of the Process Control File(s) to be checked.
- The ***PCFpathname*** will be in the ClearCase VOB if the Process Control Files are checked into ClearCase.

3 At the UNIX prompt on an AIT Sun, type **/ecs/formal/TOOLKIT/bin/sun5/pccheck.sh -i *PCFfilename* > *ResultsFile***, press **Return**.

- The ***PCFfilename*** is the full path name (directory and file name) to the Process Control File to check.
- The ***ResultsFile*** is the file name for the results that are output.
- The PCF Checker is also available on the SPR SGI machines. The easiest way to access it is to set a SDP Toolkit environment (any will do for purposes here, see Section 9.2) and type **$PGSBIN/pccheck.sh -i *PCFfilename* > *ResultsFile***, press **Return**.

4 At the UNIX prompt on the SPR SGI, **p0spg01**, type **vi *ResultsFile***, press **Return**.

- The ***ResultsFile*** is the file name for the output results as produced in step 4.
- Any text editor may be used for this procedure step.

26.5.7 Extracting Prologs

The Project standards and guidelines are contained in the latest version of the document Data Production Software and Science Computing Facility (SCF) Standards and Guidelines (423-16-01). This ESDIS document mandates that science software delivered to the DAACs to be integrated into the ECS contain prologs in the source files. Prologs are internal documentation containing information about the software. The details are specified in the ESDIS document. Prologs must be at the top of every function, subroutine, procedure, or program module.

This procedure describes using the Prolog Extractor to extract prologs into a file. Note that the prolog extractor only extracts the prologs it finds. It does not check the contents of prologs.

The following is a list of tools and/or assumptions:

1. The SSIT Manager is running.
2. Prologs are assumed to be delimited by particular delimiters depending on the language type. Delimiters are listed in the table below:

Table 26.5.7-1. Prolog Delimiters

Language	Type	Delimiter
FORTRAN 77	source	!F77
Fortran 90	source	!F90
C	source	!C
FORTRAN 77	include	!F77-INC
Fortran 90	include	!F90-INC
C	include	!C-INC
Any Language	any	!PROLOG
All Languages	The end delimiter is always !END	

- The Prolog Extractor recognizes the language type of the file by its file name extension. The table below lists assumed file name extensions:

Table 26.5.7-2. File Name Extensions

File Type	File Name Extensions
FORTRAN 77	f, f77, ftn, for, F, F77, FTN, FOR
Fortran 90	f90, F90, f, F
FORTRAN 77/Fortran 90 include	inc, INC
C	c
C/C++ header	h

Extracting Prologs

1 The Prolog Extractor can be started from the UNIX prompt. To do this, at the UNIX prompt on the AIT Sun, type **/data3/ecs/TS1/CUSTOM/bin/DPS/EcDpAtMgrPrologs**, press **Return**

or

From the SSIT Manager, select the **Tools** → **Standards Checkers** → **Prolog Extractor** from the menu.

- An xterm will be displayed on the AIT Sun.
- Select the default ConfigFile. The output goes to a file called Prologs.txt in the directory from which the SSIT Manager was started.
- The Prologs.txt file can be viewed by changing directories to the SSIT Manager directory and invoking a text editor. The file may also be sent to a printer.

2 At the **Files(S)? (-h help)** prompt, type in the file names and/or directory names containing the files.

- Separate items with spaces.
- The contents of the directory will be search recursively for files with valid file name extensions.
- Use **./** to indicate current directory.
- The time needed for the Prolog Extractor could be very long for large numbers of files and directories.
- When extraction is complete, the message **Output written to file: ./prologs.txt** will be displayed.

3 At the program prompt **Hit Enter for another, ”q <Enter>“ to quit:** , press **Enter** to repeat process with another set of source files or type **q** and press **Enter** to quit.

- The xterm will disappear.

4 At a UNIX prompt on the AIT Sun, type **vi prologs.txt,** then press the **Enter** key.

- The extracted prologs file, named **prologs.txt,** will be brought into the editor.
- The default location of the **prologs.txt** file is the directory from which the SSIT Manager was invoked.

5 Once the extracted prologs file has been examined, exit the editor.

26.6 Compiling and Linking Science Software

Science software to the DAACs is in the form of source files. In order to be run and tested within the ECS, this science software has to be compiled and linked to form the binary executables that run within the PGEs. Science software is developed at independent Science Computing Facilities (SCFs) using the SDP Toolkit. The SDP Toolkit allows science software to be developed for ECS at independent SCFs. Once delivered to the DAACs for SSI&T, science software needs to be compiled and linked to one of the SDP Toolkit versions resident at the DAAC. The (PCFs) Process Control Files provide the interface between the science software and the production system in the ECS. Since the process control files delivered to the DAACs for SSI&T were created and used at the SCFs, the path names in the PCF will need to be checked and revised to work at the DAACs.

To save time for the SSI&T Training Lesson, the compile and link with the SCF Version of the Toolkit will be omitted. The procedures are included in the student guide for future reference.

The next step is to set up a DAAC version SDP Toolkit environment, compile the PGE, and link to the DAAC Toolkit. This procedure will be performed at the SSI&T Training. The procedure steps for the two processes are the same except for the set up for the Toolkit environment and link with the corresponding Toolkit library.

26.6.1 Updating the Process Control File

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. A PCF for the PGE has been delivered and is available, accessible, and has read permissions.

To update the PCF, execute the procedure steps that follow:

1 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to the SGI.

- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then telnet to the SGI.

2 If required, at the UNIX prompt on the SGI, type **cleartool setview *ViewName***, press **Return**.

- The ***ViewName*** is the name of a view allowing the PCF to be accessible.
- This step is only necessary if the PCF is in ClearCase (in the VOB under configuration management).

3 At the UNIX prompt on the Sun or on the SGI, type **cd *PCFpathname***, press **Return**.

- The ***PCFpathname*** is the full path name to the location of the PCF. This location will be in the ClearCase VOB if the PCF is under configuration management.

4 At the UNIX prompt on the Sun or on the SGI, type **cleartool checkout -nc *PCFfilename***, press **Return**.

- The ***PCFfilename*** is the file name of the PCF that is to be checked out (and later modified). The **-nc** flag means “no comment”; it suppresses ClearCase from prompting for a comment to be associated with the check out step.

5 Run the Process Control File Checker on the delivered PCF.

- This will verify that the delivered PCF is correct before editing.

6 At a UNIX prompt on the Sun, type **vi *PCFfilename***, press **Return**.

- The ***PCFfilename*** is the file name of the PCF to update.
- Any text editor may be used such as *emacs*. For example, **emacs AST02.pcf**, press **Return**.

7 In the file, make changes to the default directories specified in each section of the PCF. All path names specified in the PCF must exist on the SGI.

- Each section begins with a line consisting of a **?** in the first column followed by a label:

```
? PRODUCT INPUT FILES
? PRODUCT OUTPUT FILES
? SUPPORT INPUT FILES
? SUPPORT OUTPUT FILES
```

```
? INTERMEDIATE INPUT
? INTERMEDIATE OUTPUT
? TEMPORARY I/O
```

- Each of the above section heading lines will then be followed (not necessarily immediately; there may be comment lines) by a line that begins with a **!** in the first column. These lines specify the default path names for each section.
- If the line reads:

  ! ~/runtime

  leave it unchanged. The tilde (~) is a symbol that represents $PGSHOME.
- If another path name is listed instead, it will probably need to be changed to a path name that exists at the DAAC on the SGI. When specifying a path name, use an absolute path name, not a relative path name.

8 In the file, look for science software specific entries in each section and make changes to the path names (field 3) as necessary. All path names specified in the PCF must exist on the SGI.

- The science software specific entries will have logical IDs (first field) *outside* of the range 10,000 to 10,999.
- Where necessary, replace the path names in the third field of each entry with the path names appropriate to the DAAC environment.
- Do not alter file entries that are used by the SDP Toolkit itself. These have logical IDs *in* the range 10,000 to 10,999.
- For example, if the following entry was found in the PCF:

  100|A.granule|/MODIS/run/input||||1

  change /MODIS/run/input to the appropriate path name in the DAAC where the file A.granule is stored.
- When specifying a path name, use an absolute path name, not a relative path name.
- Do not include the file name with the path name. The file name belongs in field 2 by itself.

9 In the file, verify that the SUPPORT OUTPUT FILES section contains an entry to the shared memory pointer file.

- Look for the entry:

  10111|ShmMem|~/runtime||||1

  The third field may be blank; this will work too.
- If this entry is not within this section, add it.

10 Once changes have been made to the PCF, save the changes and exit the editor.

- The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, press **Return**.
- For other editors, refer to that editor's documentation.

11 Again, run the Process Control File Checker on the PCF.

12 If the PCF had been checked out of ClearCase, at the UNIX prompt on the SGI, type **cleartool checkin -nc *PCFfilename***, press **Return**.

- The ***PCFfilename*** is the file name of the modified PCF. The **-nc** flag means “no comment”; it suppresses ClearCase from prompting for a comment to be associated with the check in step.

26.6.2 Setting Up an SDP Toolkit Environment

The purpose of the SDP Toolkit is to allow science software to be developed for ECS at independent SCFs and to provide:

- An interface to the ECS system, including PDPS and CSMS and information management.
- A method for Science software to be portable to different platforms at the DAAC.
- A method to reduces redundant coding at the SCF.
- Value added functionality for science software development.

The latest versions of Toolkit and accompanying Metadata tools can be found on the WWW.

Information on the SDP Toolkit, HDF-EOS, and EOSView at URL:

http://newsroom.gsfc.nasa.gov/sdptoolkit/toolkit.html

Please note: Internet links cannot be guaranteed for accuracy or currency.

The SDP Toolkit is divided into two groups of tools:

26.6.2.1 Mandatory Tools

- Error and Status Message Facility (SMF) - provides general error handling, status log messaging, and interface to CSMS services.
- Process Control Tools - provides the primary interface to the PDPS. Allows access to physical filenames and file attributes and retrieval of user defined parameters.
- Generic Input/Output - provides the means to open and close support, temporary and intermediate duration files.
- Memory Allocation Tools - simple wrappers on native C functions which track memory usage in the SDPS, and shared memory tools that enable the sharing of memory among executables within a PGE.

26.6.2.2 Optional Tools

- Ancillary Data Access - provides access to NMC data and Digital Elevation (DEM) data.
- Celestial Body Position - locates the sun, moon and the planets.
- Coordinate System Conversion - coordinate conversions between celestial references.
- Constant and Unit Conversion - physical constants and unit conversions.
- IMSL - mathematical and statistical support.

In the description of the Toolkit routines, descriptive information is presented in the following format:

TOOL TITLE

NAME:	Procedure or routine name
SYNOPSIS:	C: C language call
FORTRAN:	FORTRAN77 or Fortran90 language call
DESCRIPTION:	Cursory description of routine usage
INPUTS:	List and description of data files and parameters input to the routine
OUTPUTS:	List and description of data files and parameters output from the routine
ENTERS:	List of returned parameters indicating success, failure, etc.
EXAMPLES:	Example usage of routine
NOTES:	Detailed information about usage and assumptions
REQUIREMENTS:	Requirements from PGS Toolkit Specification, Oct. 93 which the routine satisfies

The science software delivered to the DAACs is expected to work with either the SCF SDP Toolkit or the DAAC SDP Toolkit, both of which are installed at each DAAC. During the pre-SSI&T initial testing, the SCF Toolkit should be used.

There are several versions of the SCF/DAAC SDP Toolkit installed on the SGI Power Challenges at the DAACs for the Release 4 system. The toolkit versions at the DAACs differ according to:

- Object Type - The operating system on the SGI Power Challenges on Release 4 is IRIX 6.2, a 64-bit operating system. To be backward compatible, the SGI operating system will allow new 64-bit and 32-bit objects to be built as well as the older 32-bit machines. Each of these object types is designated by placing a cc flag on the command line to enable a particular mode with the SGI C compiler.
- New 64-bit: cc flag = -64
- New 32-bit: cc flag = -n32
- Old 32-bit: cc flag = -32 (SCFs only)
- Library Type - The SDP Toolkit uses different libraries depending upon whether FORTRAN 77 or FORTRAN 90 source code is being linked. If C source code is to be linked, then either language version of the library will work.

Note - Each Toolkit library comes with a debug version, for example:
sgi32_daac_cpp/
sgi32_daac_cpp_debug/

Table 26.6.2-1 summarizes the available SDP Toolkits used by the SGI science processors.

Table 26.6.2-1. SDP Toolkits Used by the SGI Science Processors

SDP Version	Language Type	Library Object Type	$PGSBIN
SCF	C++ or C	Old 32-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi_scf_cpp
SCF	FORTRAN 77 or C	Old 32-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi_scf_f77
SCF	Fortran 90 or C	Old 32-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi_scf_f90
SCF	Thread	Old 32-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi_scf_r
SCF	C++ or C	New 32-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi32_scf_cpp
SCF	FORTRAN 77 or C	New 32-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi32_scf_f77
SCF	Fortran 90 or C	New 32-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi32_scf_f90
SCF	Thread	New 32-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi32__scf_r
SCF	C++ or C	64-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi64_scf_cpp
SCF	FORTRAN 77 or C	64-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi64_scf_f77
SCF	Fortran 90 or C	64-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi64_scf_f90
SCF	Thread	64-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi64_scf_r
DAAC	C++ or C	Old 32-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi_daac_cpp
DAAC	FORTRAN 77 or C	Old 32-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi_daac_f77
DAAC	Fortran 90 or C	Old 32-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi_daac_f90
DAAC	Thread	Old 32-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi_daac_r
DAAC	C++ or C	New 32-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi32_daac_cpp
DAAC	FORTRAN 77 or C	New 32-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi32_daac_f77
DAAC	Fortran 90 or C	New 32-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi32_daac_f90
DAAC	Thread	New 32-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi32_daac_r
DAAC	C++ or C	64-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi64_daac_cpp
DAAC	FORTRAN 77 or C	64-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi64_daac_f77
DAAC	Fortran 90 or C	64-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi64_daac_f90
DAAC	Thread	64-bit mode	$CUSTOM_HOME/TOOLKIT/toolkit/bin/sgi64_daac_r

$CUSTOM_HOME is an environment variable set to: **/usr/ecs/*MODE*/CUSTOM**
$PGSHOME is an environment variable set to: **usr/ecs/*MODE*/CUSTOM/TOOLKIT/toolkit**

The choice of which version of the SDP Toolkit to use depends upon two factors: The test being performed and the version required by the science software. For running a PGE in a simulated SCF environment (*i.e.* as if at the SCF), a SCF version of the Toolkit should be used. For running a PGE in the fully functional DAAC environment, the DAAC version should be used.

Among the DAAC versions, there are six choices. Most science software will likely require one of the 32-bit versions. If FORTRAN 77 code is being used (with or without C), then the FORTRAN 77 language version of the DAAC Toolkit must be used. Conversely, if Fortran 90 code is being used (again, with or without C), the Fortran 90 language version of the DAAC Toolkit must be used.

If both FORTRAN 77 and Fortran 90 are being used, the procedure becomes more complex. Under such circumstances, refer to document 333-EMD-001, Toolkit Users Guide for the EMD Project.

In addition to the SDP Toolkit interface to the DAAC environment, there are some other interfaces (e.g. MAPI for MODIS codes) and libraries (e.g. OCEAN library for MODIS OCEAN codes) designed to simplify the processes of building and running PGEs at DAACs. Follow the delivered documentation to build specific interfaces and libraries if necessary.

This procedure describes how to set up the appropriate SDP Toolkit environment. It involves two basic steps. First, set the SDP Toolkit home directory in the environment variable PGSHOME. The second step is to source (run) the set up script in the appropriate bin directory. This step results in a number of other environment variables getting set that will be needed.

26.6.2.3 Setting Up the SDP Toolkit Environment

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The C shell (or a derivative) is the current command shell.

To check set up a SDP Toolkit environment, execute the procedure steps that follow:

1 At the UNIX prompt on the SGI, type **setenv PGSHOME *ToolkitPathname***, press **Return**.

- The ***ToolkitPathname*** is the home directory of the particular SDP Toolkit version being used. Refer to Table 26.6.2-1. Note that the setting of PGSHOME shown in this table may differ in your local DAAC.
- Korn shell users, type **PGSHOME=*ToolkitPathname*; export PGSHOME**, press **Return**.

2 At the UNIX prompt on the SGI, type **source $PGSHOME/bin/*sgiX*/pgs-dev-env.csh**, press **Return**.

- The ***sgiX*** is one of: **sgi** for 32-bit version of the Toolkit or **sgi64** for 64-bit version of the Toolkit. Refer to the last column of Table 26.6.2-1. for path names to the file to source.
- Korn shell users, type **. $PGSHOME/bin/*sgiX*/pgs-dev-env.ksh**, press **Return** (note the "dot" and then space at the beginning of this command).

3 This step is optional. Edit the file $HOME/.cshrc and add the line **alias *aliasname* 'setenv PGSHOME *ToolkitPathname*; source $PGSHOME/bin/*sgiX*/pgs-dev-env.csh; echo "*textmessage*" '**.

- The ***aliasname*** is the name of the alias. For example, to set up an environment for the DAAC version of the Toolkit for FORTRAN 77 (or C), you might use **DAACf77** as an ***aliasname***.
- The ***ToolkitPathname*** is the home directory of the particular SDP Toolkit version being used. Refer to Table 26.6.2-1. Note that the setting of PGSHOME shown in this table may differ in your local DAAC.
- The ***sgiX*** is one of: **sgi** for 32-bit version of the Toolkit or **sgi64** for 64-bit version of the Toolkit.
- The ***textmessage*** is a message that will be echoed to the screen signifying that a new Toolkit environment has been set up. It must be enclosed within
- double quotes ("). An example may be, **"DAAC F77 Toolkit environment is now set."**
- A complete example (it should be all on one line in the .cshrc file):
  alias DAACf77 'setenv PGSHOME /$CUSTOM/TOOLKIT/bin/sgi64_daac_f77/; source $PGSHOME/bin/sgi/pgs-dev-env.csh; echo "DAAC F77 Toolkit environment is now set" '
- Other aliases for other versions of the Toolkit can be set up similarly.

26.6.2.4 An Example of Compile Procedures Used to Produce a PGE.exe

Setup for PGE07:

1 **/home/emcleod/MODIS/STORE/PGE07/MOD_PR10/source**

2 **rm MOD_PR10.exe**

3 **rm *.o**

4 **setenv PGSHOME /usr/ecs/OPS/CUSTOM/TOOLKIT/sgi32_daac_f77/**

5 **source $PGSHOME/bin/sgi32/pgs-dev-env.csh**

6 **source /home/emcleod/MODIS/STORE/PGE07/MOD_PR10/source/MODIS_setup.csh.pge07**

7 **alias**

8 **n32_f77**

9 **env**

10 **make -f MOD_PR10.mk &**

11 **ls -l *exe**

12 **setenv PGS_PC_INFO_FILE /home/emcleod/MODIS/STORE/PGE07/MOD_PR10/source/MOD_PR10.pcf**

13 **ls**

14 **MOD_PR10.exe &**

15 Confirm execution when done by looking at file : **vi MOD_PR10_ClopyL1BmetaToSnow.c**

16 See if job is running: **ps -u emcleod "time updating for MOD_PR10"**

17 **ps -u emcleod**

```
  PID TTY     TIME CMD
  267 ?       3:13 biod
25825 pts/11  0:01 csh
21994 pts/10  0:01 csh
23215 pts/16  0:00 csh
26242 pts/10  0:07 MOD_PR10.
26089 pts/11  0:01 xedit
26318 pts/10  0:00 ps
```

18 **pwd**

/tmp_mnt/home/emcleod/MODIS/STORE/PGE07/MOD_PR10/source

19 **ls**

```
MODIS_setup.csh.pge07      MOD_PR10_CopyL1BmetaToSnow.c
MODIS_setup_OPS            MOD_PR10_CopyL1BmetaToSnow.o
```

MOD_PR10.exe MOD_PR10_MakeMeta.c
MOD_PR10.h MOD_PR10_MakeMeta.o
MOD_PR10.mcf MOD_PR10_Process_Cloud.c
MOD_PR10.mk MOD_PR10_Process_Cloud.o
MOD_PR10.pcf MOD_PR10_Process_GEO.c
MOD_PR10_AAmain.c MOD_PR10_Process_GEO.o
MOD_PR10_AAmain.o MOD_PR10_Process_L1B.c
MOD_PR10_Compute_Snow.c MOD_PR10_Process_L1B.o
MOD_PR10_Compute_Snow.o MOD_PR10_Process_SnowFile.c
MOD_PR10_CopyGEOmetaToSnow.c MOD_PR10_Process_SnowFile.o
MOD_PR10_CopyGEOmetaToSnow.o compile_smf.csh

26.6.2.5 Example of a PGE Executables Tar File Insertion Script

This example was produced in Drop 4 and is provided for review only. Go to Section 26.12.13, Placing the Science Software Executable Package (SSEP) on the Data Server, which includes the Insertion of a PGE Tar file.

```
Configuration filename?  (enter for default:
../../cfg/EcDpAtInsertExeTarFile.CFG)
ECS Mode of operations?  (enter for default: OPS)
Name of PGE?  (enter for default: PGE07)
Science software version of PGE?  (enter for default: 2)
Staged filename to insert (including FULL path)?  (enter for default:
/home/emcleod/SSEP/PGE07.tar)
Associated ASCII metadata filename to insert (including FULL path)?  (enter for
default /home/emcleod/SSEP/PGE07.tar.met)
Top level shell filename within tar file?  (enter for default: PGE07.csh)
PGE07.csh
Warning: Could not open message catalog "oodce.cat"
/usr/ecs//OPS/CUSTOM/bin/DPS/EcDpAtInsertExeTarFile: Process Framework:
ConfigFile ..//cfg/EcDpAtInsertExeTarFile.CFG  ecs_mode OPS
Performing INSERT......
Retrieved for ESDT = PGEEXE the DSS UR =
UR:15:DsShSciServerUR:13:[MDC:DSSDSR
Trying to make a request to [MDC:DSSDSRV]
Trying to make a request to [MDC:DSSDSRV]
Insert to Data Server and PDPS database update successful for:
  PGE name = 'PGE07'
  Ssw version = '2'
  ESDT = 'PGEEXE'
  ESDT Version = "001"
  staged file = '/home/emcleod/SSEP/PGE07.tar'
  metadata file = '/home/emcleod/SSEP/PGE07.tar.met'
  Top level shell name = 'PGE07.csh'
Inserted at UR:
```

’UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[MDC:DSSDSRV]:14:LM:PGEEXE:94’
Hit return to run again, ’q <return>’ to quit:

26.6.3 Compiling Status Message Facility (SMF) Files

Status Message Facility (SMF) files are used by the SDP Toolkit to facilitate a status and error message handling mechanism for use in the science software and to provide a means to send log files, informational messages, and output data files to DAAC personnel or to remote users.

Science software making use of the SMF need particular header (include) files when being built and also need particular runtime message files when being run. Both the header and message files are produced by running an SMF “compiler” on a message text file. These message text files should be part of the science software delivery to the DAAC. They typically have a .t file name extension.

This procedure describes how to compile the SMF message text files to produce both the necessary include files and the necessary runtime message files.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The C shell (or a derivative) is the current command shell.

To check compile status message facility (SMF) files, execute the procedure steps that follow:

1 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then telnet to the SGI.

- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then **telnet** to the **SGI.**
- It is recommended that this procedure begin within a new command shell on the SGI.

2 If required, at the UNIX prompt on the SGI, type **cleartool setview *ViewName***, press **Return**.

- The ***ViewName*** is the name of a view allowing the SMF files to be accessible.
- This step is only necessary if any of the SMF files are in ClearCase (in the VOB under configuration management).

3 At the UNIX prompt on the SGI, type **setenv PGSHOME *ToolkitPathname***, press **Return**. Then type, source **$PGSHOME/bin/*sgiX*/pgs-dev-env.csh**, press **Return**.

- The ***ToolkitPathname*** is the home directory of the desired SDP Toolkit version.
- The ***sgiX*** refers to the appropriate processor. For example, type **source $PGSHOME/bin/sgi/pgs-dev-env.csh**, press **Return**.

4 At the UNIX prompt on the SGI, type **cd *pathname***, press **Return**.

- The ***pathname*** is the full path name to the directory containing the SMF text files.
- The SMF text files will typically have .t file name extensions.

5 At the UNIX prompt on the SGI, type **smfcompile -f *textfile*.t -lang** , press **Return**.

- The ***-lang*** is a flag that indicates for what language to compile. This flag can be one of **-c** to produce C header files and **-f77** to produce FORTRAN 77 include files. The default is for C include files. For example, type **smfcompile -f77 PGS_MODIS_39123.t**, press **Return**.
- The ***textfile*** is the file name of the SMF text file delivered with the science software.
- The SMF text files will typically have .t file name extensions.
- File names for SMF text files usually have the "seed" value used by the file as part of its file name (*e.g.* PGS_MODIS_39123.t where 39123 is the seed number).
- Only one such SMF text file can be compiled at a time; wildcards cannot be used.
- The SMF compiler may be run with the additional flags **-r** and **-i** as in, **smfcompile -f *textfile*.t -r -i**. The **-r** automatically places the runtime message file in the directory given by the environment variable PGSMSG. The **-i** automatically places the include file in the directory given by the environment variable PGSINC. For example, type **smfcompile -f90 -r -i -f PGS_MODIS_39123.t**, press **Return**. Note that the **-f** flag must always be immediately followed by the name of the text file.

6 If necessary, at the UNIX prompt on the SGI, type **mv *IncludeFilename* $PGSINC**, press **Return**. Then, type **mv *RuntimeFilename* $PGSMSG**, press **Return**.

- This step is only required if either the **-r** or the **-i** flag were not used in step 5.
- The ***IncludeFilename*** is the name of the include file created in step 5.
- The ***RuntimeFilename*** is the name of the runtime message file created in step 5.
- For example, type **mv PGS_MODIS_39123.h $PGSINC**, press **Return**. And then type, **mv PGS_MODIS_39123 $PGSMSG**, press **Return**.

26.6.4 Building Science Software with the SCF Version of the SDP Toolkit

In order to be tested at the DAAC, science software must be compiled and linked to produce binary executables. These binary executables are then packaged into one or more shell scripts as defined by the science software developer (Instrument Team). These science software packages are the Product Generation Executives (PGEs) delivered to the DAACs during SSI&T. PGEs are the smallest schedulable unit of science software in the ECS.

Building science software into PGEs should be done in accordance with supplied documentation. Such documentation should describe the process in detail. In general, science software deliveries will come with make files or other build scripts to automate the build process.

In general, science software will be built, run, and tested with the SCF version of the SDP Toolkit to ensure that the software has been successfully ported to the DAAC. Once this test has been completed successfully, the science software will be re-built, rerun, and re-tested with the DAAC version of the SDP Toolkit. Only with the DAAC Toolkit can the PGE be run within the ECS.

This procedure describes some general principals that may or may not be applicable to a particular science software delivery for building a PGE with the SCF version of the SDP Toolkit. See Section for Building a PGE with the DAAC version of the SDP Toolkit.

Building Science Software with the SCF Version of the SDP Toolkit - Activity Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The C shell (or a derivative) is the current command shell.

To build science software with the SCF version of the SDP Toolkit, be aware of the "typical" procedure steps that follow:

1 Read all instructional material supplied with the science software delivery. Such material should be the primary source of information on how to build the science software.

- Read the ***Systems Description*** document and the ***Operations Manual*.** Both of these or their equivalent should be in the delivery.
- Typically, there will be "readme" files accompanying each PGE in the directory structure, perhaps in a doc directory.
- Text files (ASCII) may be viewed with the UNIX command, ***more*** or with the ***vi* editor.**
- PostScript documents may be viewed with ***ghostview,*** which is accessible via the SSIT Manager.
- PDF formatted documents may be viewed with ***acroread,*** the Acrobat Reader, also accessible via the SSIT Manager.
- Documents in Microsoft Word and related formats may be viewed through the Microsoft Windows™ 3.1 emulator. The MS Windows emulator may be accessed from the SSIT Manager.

2 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then **telnet** to the **SGI.**

- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then **telnet** to the **SGI.**
- It is recommended that this procedure begin within a new command shell on the SGI.

3 At the UNIX prompt on the SGI, type **setenv PGSHOME *ToolkitPathname***, press **Return**. Then type, source **$PGSHOME/bin/*sgiX*/pgs-dev-env.csh**, press **Return**.

- The ***ToolkitPathname*** is the home directory of the desired SDP Toolkit version, in this case, an SCF version.

- The ***sgiX*** refers to the appropriate processor. For example, type **source $PGSHOME/bin/sgi/pgs-dev-env.csh**, press **Return**.

4 If make files are in ClearCase, at the UNIX prompt on the SGI, type **cleartool setview *ViewName***, press **Return**. Then, **cd *pathname***, press **Return**. And **cleartool checkout -nc *makefile***, press **Return**.

- The ***ViewName*** is the name of a view allowing the make files to be accessible.
- The ***pathname*** is the full path name of the directory (in the **VOB**) where the make file has been checked in.
- The ***makefile*** is the name of the make file to examine and possibly modify.
- This step is only necessary if any of the make files (or build scripts) are in **ClearCase** (in the **VOB** under configuration management).

5 Examine and alter (if necessary) any make files using any text editor **(*vi, emacs*).**

- There may be several make files for a particular **PGE**.
- Verify that compiler, compiler flag settings, and other environment variable settings are appropriate.
- The Toolkit set up (from step 3) will set many environment variables that can be used in the make files. To see the current environment variable settings, at the UNIX prompt on the **SGI**, type **env**, press **Return**.

6 Compile any required status message facility (SMF) files and place the header file(s) in the proper directory for building. Example: **p0spg01{cmshared}>/usr/ecs/TS1/CUSTOM/ssit/PGE32/message**.

7 Verify that the directory structure for the PGE source files matches the directory structure expected by the make files or build scripts.

- Deliveries may come with install scripts that place files into various directories according to some predefined structure.

8 If necessary, at the UNIX prompt on the SGI, type **cleartool checkout -nc *filename***, press **Return**.

- The ***filename*** is the file name of the **executable, object file,** or **make file** to be checked out of ClearCase. The **-nc** flag means “no comment”; it suppresses ClearCase from prompting for a comment to be associated with the check out step.
- Note that checking in executable or object files is ***not*** recommended in the first place.

9 Build the software in accordance with instructions delivered.

- Science software deliveries may come with a single, top-level script to do the entire build or the build process could involve a series of steps, each of which should be described fully in the delivered documentation.
- Choose the most appropriate optimization/debugger flag. During testing, the **"-g"** is often used. This results in larger and slower executables, but assists in debugging. For

production, the **"-O"** flag may be used to optimize execution time. Variants of the **"-g"** and **"-O"** flags may be incompatible.

10 If necessary, at the UNIX prompt on the SGI, type **cleartool checkin *filename* -nc**, press **Return**.

- The ***filename*** is the file name of the executable, object file, or make file to be checked into ClearCase. The **-nc** flag means "no comment"; it suppresses ClearCase from prompting for a comment to be associated with the check in step.
- Note that checking in executable or object files is ***not*** recommended.

26.6.5 Building Science Software with the DAAC Version of the SDP Toolkit

In general, science software will be built, run, and tested with the SCF version of the SDP Toolkit to ensure that the software has been successfully ported to the DAAC. Once this test has been completed successfully, the science software will be re-built, rerun, and re-tested with the DAAC version of the SDP Toolkit. Only with the DAAC Toolkit can the PGE be run within the ECS.

This procedure describes some general principals that may or may not be applicable to a particular science software delivery for building a PGE with the DAAC version of the SDP Toolkit.

Building Science Software with the DAAC Version of the SDP Toolkit - Activity Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The C shell (or a derivative) is the current command shell.

To build science software with the DAAC version of the SDP Toolkit, be aware of the "typical" procedure steps that follow:

1 Read all instructional material supplied with the science software delivery. Such material should be the primary source of information on how to build the science software.

- Read the ***Systems Description*** document and the ***Operations Manual***. Both of these or their equivalent should be in the delivery.
- Typically, there will be **"readme"** files accompanying each PGE in the directory structure, perhaps in a doc directory.
- Text files (ASCII) may be viewed with the UNIX command, ***more*** or with the ***vi* editor.**
- PostScript documents may be viewed with ***ghostview,*** which is accessible via the SSIT Manager.
- PDF formatted documents may be viewed with ***acroread***, the Acrobat Reader, also accessible via the SSIT Manager.

- Documents in Microsoft Word and related formats may be viewed through the Microsoft Windows™ 3.1 emulator. The MS Windows emulator may be accessed from the SSIT Manager.

2 From the SSIT Manager, click on the **Tools** menu, then choose **Xterm**. Then **telnet** to the **SGI**.

- Alternatively, in any currently available xterm window, spawn a new session: type **xterm &**, press **Return**. Then telnet to the SGI.
- It is recommended that this procedure begin within a new command shell on the SGI.

3 At the UNIX prompt on the SGI, type **setenv PGSHOME *ToolkitPathname***, press **Return**. Then type, source **$PGSHOME/bin/*sgiX*/pgs-dev-env.csh**, press **Return**.

- The ***ToolkitPathname*** is the home directory of the desired SDP Toolkit version, in this case, a DAAC version.
- The ***sgiX*** refers to the appropriate processor. For example, type **source $PGSHOME/bin/sgi/pgs-dev-env.csh**, press **Return**.

4 If make files are in ClearCase, at the UNIX prompt on the SGI, type **cleartool setview *ViewName***, press **Return**. Then, **cd *pathname***, press **Return**. And **cleartool checkout -nc *makefile***, press **Return**.

- The ***ViewName*** is the name of a view allowing the make files to be accessible.
- The ***pathname*** is the full path name of the directory (in the VOB) where the make file has been checked in.
- The ***makefile*** is the name of the make file to examine and possibly modify.
- This step is only necessary if any of the make files (or build scripts) are in **ClearCase (**in the **VOB** under configuration management).

5 Examine and alter (if necessary) any make files using any text editor (*vi, emacs*). If the software had already been built and tested with the SCF version of the SDP Toolkit, this step may be unnecessary.

- There may be several make files for a particular PGE.
- Verify that compiler, compiler flag settings, and other environment variable settings are appropriate.
- The Toolkit set up (from step 3) will set many environment variables that can be used in the **make** files. To see the current environment variable settings, at the UNIX prompt on the **SGI**, type **env**, press **Return**.

6 Compile any required status message facility (SMF) files and place the header file(s) in the proper directory for building.

7 Verify that the directory structure for the **PGE** source files matches the directory structure expected by the make files or build scripts.

- Deliveries may come with install scripts that place files into various directories according to some predefined structure.

8 If necessary, at the UNIX prompt on the SGI, type **cleartool checkout -nc *filename***, press **Return**.

- The ***filename*** is the file name of the executable, object file, or make file to be checked out of ClearCase. The **-nc** flag means “no comment”; it suppresses ClearCase from prompting for a comment to be associated with the check out step.
- Note that checking in executable or object files is ***not*** recommended in the first place.

9 Build the software in accordance with instructions delivered.

- Science software deliveries may come with a single, top-level script to do the entire build or the build process could involve a series of steps, each of which should be described fully in the delivered documentation.
- Choose the most appropriate optimization/debugger flag. During testing, the **"-g"** is often used. This results in larger and slower executables, but assists in debugging. For production, the **"-O"** flag may be used to optimize execution time. Variants of the **"-g"** and **"-O"** flags may be incompatible.

10 If necessary, at the UNIX prompt on the **SGI,** type **cleartool checkin *filename* -nc**, press **Return**.

- The ***filename*** is the file name of the executable, object file, or make file to be checked into ClearCase. The **-nc** flag means “no comment”; it suppresses ClearCase from prompting for a comment to be associated with the check in step.
- Note that checking in executable or object files is ***not*** recommended.

26.6.6 PGE Checkout

The following Servers/Services must be up and operational:

NONE.

The following must have occurred between those Servers/Services:

NONE.

What the user must do before trying SSIT functionality:

In normal SSIT (at the DAACs) the DAP would be untarred, and the source code recompiled and tested.

What must be done via SSIT tools:

Since SSIT is just a calibration of various tools, there is no specific order for which they must be run. All tools can be started from the SSIT Manager and can be executed on their own.

For SparcWorks (for code analysis and debugging), choose Tools menu and then Code Analysis submenu. See SparcWorks manuals for SparcWorks operation.

For various office tools, choose Tools menu and the Office Automation submenu. Choose from MS Windows (a simulator to allow the user to run Windows programs), Ghostview (a viewer), Netscape (for web access), Acrobat (for document viewing), and DDTS (for problem reporting).

For Standards checkers, choose Tools and then the Standards Checkers submenu.

FORCHECK is a COTS Fortran language checking program.

The Prohibited Function Checker will examine source code for functions that are not permitted On Prohibited Function GUI, choose Analyze to select files to examine. Hit the Ok button once selections are made and a message at the top of the GUI will indicate if prohibited functions have been found. If prohibited functions HAVE been found, use the View button to view the source code with the prohibited call. The Help button gives further information on how to work the GUI.

The Process Control File Checker examines selected PCFs and highlights any errors. The Process Control File GUI allows the user to work through the directory structure on the local machine and select PCFs to be checked. Click the Check PCF button to check a selected PCF. Again, the Help button provides more information.

26.7 Running a PGE in a Simulated SCF Environment

Science software delivered to the DAACs for SSI&T was developed and tested at individual SCFs using the SCF version of the SDP Toolkit. Before linking the software with the DAAC version of the Toolkit and integrating it with the ECS, it is prudent to first link the software to the SCF version of the Toolkit and run it as it was run at the SCF. This type of testing can reveal problems associated with the process of porting the software to another platform whose architecture may be quite different from the one on which the software was developed.

A simulated SCF environment means that the software is built using the SCF version of the Toolkit and is run from the UNIX command line. The Planning and Data Processing System (PDPS) and the Data Server are not involved.

The procedures that follow describe how to run the science software in a simulated SCF environment.

26.7.1 Setting Up the Environment for Running the PGE

Running a PGE that has been built with the SCF version of the SDP Toolkit requires some environment set up as it does at the SCF. This procedure describes how to set up a simulated SCF environment.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The Process Control File (PCF) exists and has been tailored for the DAAC environment.
2. The C shell or a derivative (*e.g.* T shell) is the current user shell.

To set up an environment for running the PGE, execute the procedure steps that follow:

1 Open an **xterm** window and then **telnet** to the **SPR SGI**.

- It is recommended that this procedure begin within a new command shell on the SPR SGI.

2 At the UNIX prompt on the SPR SGI, type **setenv PGSHOME *ToolkitPathname***, press **Return**. Then type, **source $PGSHOME/bin/*sgiX*/pgs-dev-env.csh**, press **Return**.

- The ***ToolkitPathname*** is the home directory of the desired SDP Toolkit version, in this case, an SCF version. For example, **p0spg01{cmshared}>setenv PGSHOME /usr/ecs/TS1/CUSTOM/TOOLKIT/toolkit/bin/sgi64_daac_f90.**
- The ***sgiX*** refers to the appropriate processor. For example, **p0spg01{cmshared}>source $PGSHOME/pgs_dev_env.csh**, press **Return**.

3 At the UNIX prompt on the SPR SGI, type **setenv PGS_PC_INFO_FILE *PCFpathname*/*PCFfilename***, press **Return**.

- The ***PCFpathname*** is the full path name to the location of the Process Control File (PCF) to be associated with this PGE.
- The ***PCFfilename*** is the file name of the PCF.
- For example, **setenv PGS_PC_INFO_FILE /usr/ecs/TS1/CUSTOM/ssit/PGE32/PCF/PGE32.pcf**, press **Return**.

4 This step is optional. At the UNIX prompt on the SPR SGI, type **rm *LogPathname*/*LogFilename***, press **Return**.

- The ***LogPathname*** is the full path name to the location of the PGE log files for this PGE.
- The ***LogFilename*** is the file name of the PGE log file to remove from a previous run of the same PGE. PGE log files can be Status, User, or Report.
- The ***LogFilename*** may use wildcard characters to remove all of the log files at the same time.
- This step is optional. If log files from a previous run of the same PGE are not removed, they will be appended with the information from the current run.
- The environment will then be set up. Continue on to Section 12.2.

5 If necessary, set any other shell environment variables needed by the PGE by sourcing the appropriate scripts or setting them on the command line.

- For example, for a PGE requiring IMSL, at the UNIX prompt on the SPR SGI, type **source /usr/ecs/<mode>/COTS/imsl/vni/ipt/bin/iptsetup.cs**, press **Return**.

- For some PGEs, the environment variables to be set will be specified in the documentation or the files to source will be supplied in the delivery. Refer to documentation included in the delivery.

26.7.2 Running and Profiling the PGE

Profiling a PGE refers to the process of gathering information about the runtime behavior of a PGE. The information includes the wall clock time, user time and system time devoted to the PGE; the amount of memory used; the number of page faults; and the number of input and output blocks.

The Planning and Data Processing System (PDPS) database must be populated with the above information when the PGE is registered with the PDPS during the integration phase of SSI&T. This information may be delivered with the PGE or it may need to be determined at the DAAC during SSI&T. This procedure addresses the latter need.

Note that profiling, as used here, does not involve altering the binary executable to produce instrumented code.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The PGE has been built successfully with the SCF version of the SDP Toolkit .
2. The required SMF runtime message files have been produced and placed in the correct locations.
3. The Process Control File (PCF) exists and has been tailored for the DAAC environment.
4. The required environment for running the PGE has been set up.
5. The required input files are available and accessible.
6. The C shell or a derivative (*e.g.* T shell) is the current user shell.

To run and profile the PGE, execute the procedure steps that follow:

1 At the UNIX prompt on the SPR SGI in the window containing the set up environment, type **cd *PGEbinPathname***, press **Return**.

- The ***PGEbinPathname*** is the full path name of the directory containing the built PGE binary executable. For example, **p0spg01{cmshared}cd /usr/ecs/TS1/CUSTOM/ssit/PGE32/bin/**, press **Return**.

2 At the UNIX prompt on the SPR SGI, type **/usr/ecs/<mode>/CUSTOM/bin/DPS/E*cDpPrR*usage *PGE.exe* >& *ResultsOut***, press **Return**.

- The ***PGE.exe*** is the name given to the PGE binary executable.

- The ***ResultsOut*** is the file name in which to capture the profiling results as well as any messages from standard output (stdout) and standard error (stderr) that may be produced by the running PGE. Note that PGEs should *not* write to stdout or stderr.
- For example:
  p0spg01{cmshared}>/usr/ecs/TS1/CUSTOM/bin/DPS/EcDpPrRusage run_PGE32.exe >& ../logs/PGE32.profile.out
- The **EcDpPrRusage** is the profiling program that outputs information about the runtime behavior of the PGE.
- Depending upon the PGE, it may take some time before the UNIX prompt returns.

3 At the UNIX prompt on the SPR SGI, type **echo $status**, press **Return**.

- The **$status** is an environment variable that stores the exit status of the previous program run, in this case, the PGE.
- A status of zero indicates success; a status of non-zero indicates an error of some kind.
- The meaning of a non-zero exit status should be documented and included with the DAPs.
- This command must be run ***immediately*** after the **EcDpPrRusage** command.

4 At the UNIX prompt on the SPR SGI, type **vi *ResultsOut***, press **Return**.

- The ***ResultsOut*** is the file name under which the profiling output was saved. Other output of the PGE may also be in this file.
- The **EcDpPrRusage** results may then be recorded and used when the PGE is registered in the PDPS.
- Any text editor/viewer may be used.
- **Sample of an Rusage File produced:**

```
p0spg01{emcleod}6: more Profile.out
#     source .cshrc
#     cd TEST/MOD*
#     ls
#     /usr/ecs/OPS/CUSTOM/bin/DPS/EcDpPrRusage MOD_PR10.exe >
Profile.out
p0spg01{emcleod}9: more profile.out
# Resource Usage Information
COMMAND=MOD_PR10.exe
EXIT_STATUS=0
ELAPSED_TIME=233.583145
USER_TIME=10.046158
SYSTEM_TIME=7.555547
MAXIMUM_RESIDENT_SET_SIZE=4080
AVERAGE_SHARED_TEXT_SIZE=0
AVERAGE_UNSHARED_DATA_SIZE=0
AVERAGE_UNSHARED_STACK_SIZE=0
```

PAGE_RECLAIMS=151
PAGE_FAULTS=0
SWAPS=0
BLOCK_INPUT_OPERATIONS=2
BLOCK_OUTPUT_OPERATIONS=2710
MESSAGES_SENT=0
MESSAGES_RECEIVED=0
SIGNALS_RECEIVED=0
VOLUNTARY_CONTEXT_SWITCHES=1095
INVOLUNTARY_CONTEXT_SWITCHES=2

26.7.3 Checking the PGE for Memory Leaks

One of the important types of "code inspection" is to check the PGE for memory leak. This session assumes that PGE build and command line run both been successfully run. The task requires building and running with a memory error-checking utility (such as Purify).

Integration and testing are conducted using the Goddard Automated Tools for Enhanced SSI&T (GATES) procedures. The GATES procedures are available on the web at:

http://gsfcsrvr8.gsfcmo.ecs.nasa.gov/GATES/TEST/AIRS/AIRS_FauxSwatter_EnterInfo.html.TEST

26.8 File Comparison and Data Visualization

The purpose of File Comparison is to verify that the output files produced at the DAAC are identical (within tolerances) to the test output files delivered with the DAPs. A successful comparison is a strong indication that the porting of the science software from the development facility at the SCF to the operational facility at the DAAC has not introduced any errors.

A number of file comparison tools are available during SSI&T via the SSIT Manager GUI or they can be invoked from the UNIX command line. Two tools are available for comparing HDF or HDF-EOS files, one tool for comparing ASCII files, and another tool for assisting in comparing binary files.

26.8.1 Using the GUI HDF File Comparison GUI

The following is a list of tools and/or assumptions:

1. The SSIT Manager is running.
2. Two HDF or HDF-EOS files exist with similar structures.
3. The Instrument Team has delivered test output files.
4. If either of the two HDF/HDF-EOS files is in the ClearCase VOB, a ClearCase view was set before the SSIT Manager was started.

Comparing Two HDF or HDF-EOS Files Using the HDF File Comparison GUI

1 From the SSIT Manager, select **Tools→ Product Examination → HDF** from the menu.

- The HDF File Comparison GUI window will be displayed.

2 In the HDF File Comparison Tool GUI, click on the **File 1** button.

- Read the Systems Description document and the Operations Manual. Both of these or their equivalent should be in the delivery.

26.8.2 Using the hdiff HDF File Comparison Tool

The hdiff File Comparison Tool is a text-oriented tool run from the command line. It allows comparison of two HDF or HDF-EOS files.

Comparing two HDF or HDF-EOS Files Using the hdiff File Comparison Tool

1 From the SSIT Manager, select **Tools→ Product Examination → File Comparison → HDF** from the menu.

- The HDF File Comparison Tool window will be displayed.
- An xterm window running ***hdiff*** will be displayed.

2 In the xterm window at the prompt **Options? (-h for help)**, type in any desired options then press the **Enter** key.

- To see the list of available options, type **-h** then press the **Enter** key. to the prompt.

3 In xterm window at the prompt **1st file to compare?**, type ***filename1***, then press the **Enter** key.

- The ***filename1*** is the file name of the first of two HDF or HDF-EOS files to be compared.
- If ***filename1*** is not in the current directory (the directory from which the SSIT Manage was run), include the full path name with the file name.

4 In xterm window at the prompt **2nd file to compare?**, type ***filename2***, then press the **Enter** key.

- The ***filename2*** is the file name of the second of two HDF or HDF-EOS files to be compared. Select another file.
- If ***filename2*** is not in the current directory (the directory from which the SSIT Manage was run), include the full path name with the file name. The two files will be compared and the output will be displayed in the xterm window.
- **Note: hdiff** can also be invoked from command line. This done by executing the command **/usr/ecs/<*mode* >/CUSTOM/bin/DPS/hdiff *filename1 filename2***

26.8.3 Using the ASCII File Comparison Tool

Most output files (products) from PGEs run in the DAAC will be in HDF-EOS format. A small minority may be in ASCII (text) format. The ASCII File Comparison Tool is a front-end to ***xdiff*** UNIX X Window tool for comparing two ASCII files.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

The following is a list of tools and/or assumptions:

1. The SSIT Manager is running.
2. Two ASCII files exist and have read permissions.
3. The instrument Team has delivered test output files.
4. If either of the two ASCII files is in the ClearCase VOB, a ClearCase view was set before the SSIT Manager was started.

Comparing Two ASCII Files

1 From the SSIT Manager, select **Tools→ Product Examination → File Comparison → ASCII** from the menu.
- An xterm window running *xdiff* will be displayed.

2 In xterm window at the prompt **1st file to compare?**, type ***filename1***, then press the **Enter** key. Select a descriptor or mcf file in the directory with the PGE.
- The ***filename1*** is the file name of the first of two ASCII files to be compared.
- If ***filename1*** is not in the current directory (the directory from which the SSIT Manage was run), include the full path name with the file name.

3 In xterm window at the prompt **2nd file to compare?**, type ***filename2***, then press the **Enter** key.
- The ***filename2*** is the file name of the second of two ASCII files to be compared. Select another corresponding file.
- If ***filename2*** is not in the current directory (the directory from which the SSIT Manager was run), include the full path name with the file name.
- A window labeled **xdiff** will be displayed.

4 In the window labeled **xdiff**, view the differences between the two files displayed.
- File ***filename1*** will be displayed on the left side of the window. File ***filename2*** will be displayed on the right.
- Only sections of file in which there are differences will be displayed. A “bang” character (!) at the beginning of a line indicates that a difference was found.

- For further help on *xdiff*, type **man xdiff**, in an xterm window then press the **Enter** key.
- Close the display window by using the pull down menu from the X window in the upper left corner.

5 In the xterm window at the prompt **Hit Enter for another diff, ‘q <Enter>’ to quit:**, type **q**, press **Enter** to quit, or just press **Enter** to perform another comparison.
- **Note: xdiff** can also be invoked from command line. This done by executing the command **/usr/ecs/<*mode* >/CUSTOM/bin/DPS/xdiff *filename1 filename2***

26.8.4 Using the Binary File Difference Assistant

Most output files (products) from PGEs run in the DAAC will be in HDF-EOS format. A small minority may be in some binary format. The Binary File Difference Assistant aids the user in constructing code that allows comparison of binary output files. Since there is an unwieldy number of possibilities for binary file formats, this tool cannot compare two binary files without some custom code written at the DAAC, hence, the “Assistant” in the name. The Binary File Difference Assistant aids the user by generating a makefile, a driver module, and a template comparison module in C, FORTRAN 77 or IDL (Interactive Data Language). The user then edits these templates to read the particular binary format in question according to an SCF-supplied format specification.

The binary file comparison will not be performed during the SSIT training lesson.

The following is a list of tools and/or assumptions:

1. The SSIT Manager is running.
2. Two Binary files exist and have read permissions.
3. The instrument Team has delivered test output files.
4. If either of the two Binary files is in the ClearCase VOB, a ClearCase view was set before the SSIT Manager was started.

Comparing two Binary Files

1 From the SSIT Manager, select **Tools→ Product Examination → File Comparison → Binary** from the menu.
- The Binary File Difference Assistant tool GUI will be displayed.

2 In the Binary File Difference Assistant tool GUI, click on one of the languages listed under the **Select Language** label. The choices are C, FORTRAN, or IDL.
- The choice of language depends largely on preference. It does not necessarily have to be the language that was used to create the files being compared.

3 Optionally, click on either the **Image** button or the **Structure** button located under the label **Compare Function**.

- Clicking on the **Image** button will display a code example for comparing binary files containing images.
- Clicking on the **Structure** button will display a code example for comparing binary files containing structures or records.
- The displayed listing well documented and should be read.
- The language of the code will depend on the language selection made in step 2.

4 Optionally, click on either the **Image** button or the **Structure** button located under the label **Driver**.

- Clicking on the **Image** button will display a code example for a driver invoking the compare function for binary files containing images.
- Clicking on the **Structure** button will display a code example for a driver invoking the compare function for binary files containing structures or records.
- The displayed listing well documented and should be read.
- The language of the code will depend on the language selection made in step 2.

5 Optionally, click on either the **Help** button.

- A Help window will be displayed.
- To end help, click on the **Dismiss** button.
- The Help window may remain displayed while using the Binary File Difference Assistant.

6 Once familiar with the code examples (steps 3 and 4), click on the **Copy** button.

- A window labeled **Enter Unique ID** will be displayed.
- In the field labeled **Enter unique file identifier:**, type ***fileID***, click on the **OK** button.
- The ***fileID*** will be used in the file names of the files copied over. These files will be:

C:

DaacBinDiff_***fileID***.c	Compare function
DaacBinDiff_***fileID***_driver.c	Driver
DaacBinDiff_***fileID***.mak	Makefile

FORTRAN:

DaacBinDiff_***fileID***.f	Compare function
DaacBinDiff_***fileID***_driver.f	Driver
DaacBinDiff_***fileID***.mak	Makefile

IDL:

DaacBinDiff_***fileID***.pro	Compare function

DaacBinDiff_***fileID***_driver.pro	Driver
DaacBinDiff_***fileID***.sh	Shell script with here document

- The files will be copied into the directory from which the SSIT Manager is being run.

NOTE: 1. Using any desired text editor, customize the files for the job at hand. Then build the executable using the customized makefile provided (for C and FORTRAN). Then run the program to perform the binary file comparison.

2. These files are templates that can be used to assist SSIT personnel in designing customized code to perform the binary file comparison.

26.8.5 Data Visualization

In order to view the success of science software in producing scientifically valid data sets, the data needs to be displayed in forms that convey the most information. Data visualization enables this to be done.

There are two visualization tools provided to the DAAC: EOSView and Interactive Data Language (IDL). These tools are both accessible via the SSIT Manager. EOSView is user friendly GUI for creating two-dimensional displays from HDF-EOS objects(Grid, Swath) as well as the standard HDF objects (SDS, Vdata, Image, Text). It has additional features such as thumbnail-panning, colorization, zooming, plotting, and animation. Only some aspects of data visualization will be addressed in this training material. For further information, see the related references.

IDL is a COTS display and analysis tool widely applied in the scientific community, It is used to create two-dimensional, three dimensional (volumetric), and surface/terrain displays from binary, ASCII, and many other formats in addition to HDF.

26.8.5.1 Data Visualization with EOSView

26.8.5.1.1 Viewing Product Metadata with the EOSView Tool

This procedure describes how to use the EOSView tool to inspect the metadata in the HDF-EOS output file from a PGE. To view product metadata with the EOSView tool, execute the procedure steps that follow:

Log into an Algorithm and Test Tools (AITTL) environment using a machine so configured. At the **PVC** this machine is **p0ais01**

1 Telnet into **p0ais01.**

2 logon using your own ID and Password

3 **cd /usr/ecs/TS1/CUSTOM/eosview**

4 Select **EOSView**

- The EOSView GUI will be displayed.

5 Use the select buttons to guide you toward the view desired.

The following is a list of tools and/or assumptions:

1. The SSIT Manager is running.
2. The output file is HDF-EOS and has been created and populated with metadata using the SDP Toolkit metadata tools.

Viewing Product Metadata

1 From the SSIT Manager, select **Tools**→ **Product Examination** → **EOSView** from the menu.

- The EOSView GUI will be displayed.

2 In the GUI labeled **EOSView - EOSView Main Window**, click on the **File** menu and select **Open.** The **Filter** GUI will be displayed.

3 In the **Filter** subwindow, enter full path name and file name wildcard template. For example, enter ***/home/MyDirectory/MySubdirectory/****.

- The ***/home/MyDirectory/MySubdirectory/**** represents the location to the directory containing the HDF-EOS files to examine.
- The asterisk (*) is a wildcard template that represents all files in that directory; other wildcard templates can narrow the search further, *e.g.* ***.hdf**.
- Use the **Directories** field to further select the correct directory.
- Files found matching the wildcard template in the chosen directory will be displayed in **Files** subwindow.

4 In the **Files** subwindow, click on the file name of the HDF-EOS file to examine. Then click on the **OK** button.

- A GUI labeled **EOSView - *MyOutputFile.hdf*** will be displayed where *MyOutputFile.hdf* is the file name of the file chosen in step 3.
- Be patient - this GUI may take some time to appear, particularly for large files.
- Once displayed, a list of HDF objects will appear in the main window. If nothing is listed, it means that no HDF objects were found within the file.

5 In the GUI labeled **EOSView - *MyOutputFile.hdf***, double click on an object listed for which data is to be inspected.

6 Go to procedure depending upon the type of the object selected in step 5.

- Proceed to Section 15.2.1 if object is an HDF Image.
- Proceed to Section 15.2.2 if object is an HDF-EOS Grid.

- Proceed to Section 15.2.3 if object is an HDF-EOS Swath.
- Proceed to Section 15.2.4 if object is an HDF SDS.
- Proceed to Section 15.2.5 is object is in another format.

7 Repeat steps 5 and 6 for each HDF object within the selected HDF-EOS file for which data is to be examined.

8 In the GUI labeled **EOSView - *MyOutputFile.hdf***, click on the **File** menu and select **Close**.
- The **EOSView - *MyOutputFile.hdf*** GUI will disappear.
- Be patient - this GUI may take some time to disappear, particularly for large files.

9 In the GUI labeled **EOSView - EOSView Main Window**, click on the **File** menu and select **Exit**.

28.8.5.1.2 Viewing HDF Image Objects

This procedure describes how to use the EOSView tool to view science Images (typically, browse images) in the HDF-EOS output file from a PGE. See Section 15.2 for how to select an HDF Image object within an HDF-EOS file. An HDF-EOS file is defined as a file produced using the SDP Toolkit metadata tools and having, at a minimum, the ECS core metadata.

Once an HDF Image is displayed, a number of options are available. These include colorization, zooming, panning, and animation. Each is described in the procedures below as an optional step.

EOSView will display and HDF Image in its referenced default color palette if the file contains one. If the display looks “blacked out”, it may mean that no default color palette is available or referenced by the Image object. In this case, a palette will have to be selected by the user.

Zooming allows both zoom in and zoom out according to a Bilinear Interpolation or Nearest Neighbor resampling method. Bilinear Interpolation involves averaging to produce a “smoothed” display appropriate for gray scale band Images or derived geophysical parameter Images (*e.g.* temperature, vegetation index, albedo). Nearest Neighbor produces a non-smoothed display appropriate for derived thematic image-maps (*e.g.* land cover, masks).

Panning allows user to select the portion of a zoomed Image to be displayed.

Animation allows multiple Images to be displayed in an animated fashion in a number of modes. Stop-at-End mode allows Images to be displayed to the end just once. Continuous Run mode lets the animation run through the Images repeatedly and Bounce mode does a forward/reverse run through any set of Images repeatedly. All animation runs can be stopped at any time and then resumed in either the forward or reverse directions. The speed of an animation can be adjusted as well.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The output file is HDF-EOS and has been created and populated with metadata using the SDP Toolkit metadata tools and that at least one object is an HDF image (RIS8, RIS24, *i.e.* Browse data).

2. EOSView has been properly installed and is accessible to the user.

3. The HDF-EOS file has been read into EOSView by the procedures described in Section 15.2.

To view an HDF-EOS Image object with the EOSView tool, execute the procedure steps that follow:

1 In the GUI labeled **EOSView - *MyOutputFile.hdf***, double click on an Image object listed for which data is to be inspected.

- A GUI labeled **EOSView - Image Display Window - *MyImageObject*** will be displayed where *MyImageObject* will be replaced by the name of the object selected as listed. For example, **EOSView - Image Display Window - Image [512x512] ref=2 (palette)**.
- Be patient - this GUI may take some time to appear, particularly for large files.

2 Optional colorization. In the GUI labeled **EOSView - Image Display Window - *MyImageObject***, click on the **Palette** menu, then select one of the palettes listed: **Default**, **Greyscale**, **Antarctica**, **Rainbow**, or **World Colors**.

- The selection of palette will not result in any change to the data in the file or to the object's default palette.
- This selection may be repeated until the desired palette is chosen.

3 Optional zooming. In the GUI labeled **EOSView - Image Display Window - *MyImageObject***, click on the **Zooming** menu, then select **Select** and then select one of the resampling methods listed: **Bilinear Interpolation** or **Nearest Neighbor**. Then click on the **Zoom In** or **Zoom Out** buttons to apply the method.

- The selection of re-sampling method and zoom will not result in any change to the data in the file.
- The zooming options may be repeated as desired.

4 Optional panning while zooming. In the GUI labeled **EOSView - Image Display Window - *MyImageObject***, a thumbnail representation of the entire Image will be displayed in the subwindow labeled **Pan Window**. A hollow rectangle on the thumbnail indicates the portion of the Image that is being displayed in the main window. Use the mouse left button to click and drag the rectangle to a new location on the thumbnail image.

- The portion of the zoomed Image shown in the main window will be the portion indicated by the hollow rectangle on the thumbnail image.
- The panning will not result in any change to the data in the file.
- The panning option may be repeated as desired.

5 To end the session with colorization, zooming, or panning, in the GUI labeled **EOSView - Image Display Window - *MyImageObject***, click on the **File** menu and select **Close**.

- The **EOSView - Image Display Window - *MyImageObject*** GUI will disappear.

6 Optional animation. In the GUI labeled **EOSView - *MyOutputFile.hdf***, click on the **Options** menu, then select **Animated images**.

- A GUI labeled **EOSView - Image Animation Window - *MyOutputFile.hdf*** will be displayed.
- Be patient - this GUI may take some time to appear, particularly for large files.
- Optionally, click on the **Palette** menu to select a palette.
- Optionally, click on the **Options** menu and then select **Mode** to select how the animation is to be run. Choose **Stop at end**, **Continuous run**, or **Bounce**.
- Click on the Stop button, denoted by the **||** symbol (center) to halt the animation.
- Resume the animation by clicking on either the Forward Play (denoted by the **>>** symbol) or the Reverse Play (denoted by the **<<** symbol).
- There is also Forward Increment (**>|**) and Reverse Increment (**|<**) button.
- The animation speed may be adjusted by moving the **Speed** slider in either the "**+**" or "**-**" direction.
- To end animation session, click on the **File** menu and then select **Close**.

26.8.5.1.3 Viewing HDF-EOS Grid Objects

This procedure describes how to use the EOSView tool to view science data in the HDF-EOS output file that are in HDF-EOS Grid format. These are generally the science data and not browse images.

The following is a list of tools and/or assumptions:

1. The SSIT Manager is running.
2. The output file is HDF-EOS and has been created and populated with metadata using the SDP Toolkit metadata tools.
3. At least one object is an HDF-EOS Grid.

Viewing HDF-EOS Grid Objects

1 From the SSIT Manager, select **Tools→ Product Examination → EOSView** from the menu.

- The EOSView GUI will be displayed.

2 In the GUI labeled **EOSView - *MyOutputFile.hdf***, double click on a Grid object listed for which data is to be inspected. A GUI labeled **EOSView - Grid Select** will be displayed.

- Information on **Grid Information, Projection Information, Dimensions,** and **Attributes** for the selected object can be displayed by clicking on the appropriate checkboxes.

3 In the GUI labeled **EOSView - Grid Select**, click on the **Data Fields** checkbox and then click on the **OK** button. Then double click on one of the data fields listed.

- A GUI labeled **EOSView - Grid - *GridObjectName* - Start/Stride/Edge** will be displayed where *GridObjectName* will be replaced by the name of the Grid object selected in step 1.

4 To display the data in the form of a table of values, in the GUI labeled **EOSView - Grid - *GridObjectName* - Start/Stride/Edge**, click on the checkboxes for both **YDim** and **XDim** and then click on the **OK** button.

- A GUI labeled ***MyDataField*** will be displayed where ***MyDataField*** will be replaced by the name of the data field selected in step 2.

5 In the GUI labeled ***MyDataField***, click on the **File** menu and then select **Make Image**. Optionally adjust the default minimum and maximum data values and then click on the **Continue** button.

- Adjusting the minimum and maximum data values will affect only the display (contrast) and not the actual data in the file.
- A GUI labeled **EOSView - Swath/Grid Image** will appear, displaying the data field in image form.

6 Optional colorization. In the GUI labeled **EOSView - Swath/Grid Image**, click on the **Palette** menu, then select **Select** and then select one of the palettes listed: **Default**, **Greyscale**, **Antarctica**, **Rainbow**, or **World Colors**.

- The selection of palette will not result in any change to the data in the file or to the object's default palette.
- This selection may be repeated until the desired palette is chosen.

7 Optional zooming. In the GUI labeled **EOSView - Swath/Grid Image**, click on the **Zooming** menu, then select **Select** and then select one of the resampling methods listed: **Bilinear Interpolation** or **Nearest Neighbor**. Then click on the **Zoom In** or **Zoom Out** buttons to apply the method.

- The selection of resampling method and zoom will not result in any change to the data in the file.
- The zooming options may be repeated as desired.

8 Optional panning while zooming. In the GUI labeled **EOSView - Swath/Grid Image**, a thumbnail representation of the entire Image will be displayed in the subwindow labeled **Pan Window**. A hollow rectangle on the thumbnail indicates that portion of the Image that is being displayed in the main window. Use the mouse left button to click and drag the rectangle to a new location on the thumbnail image.

- The portion of the zoomed Image shown in the main window will be the portion indicated by the hollow rectangle on the thumbnail image.
- The panning will not result in any change to the data in the file.

- The panning option may be repeated as desired.

9 Optional flat file output. From the GUI ***EOSView—Grid Select***, select **Grid Dimensions** and take note of the PixelsXTrack value (the number of elements in the Grid object's raster); you will need to remember this value in order to make later use of your output flat file (since the flat file by definition contains no structural metadata). In the GUI labeled ***MyDataField***, click on the **File** menu and then select **Save**. Then select either **Binary** or **ASCII** flat file type, and assign the output file a name.

- This procedure will create an output flat file (which may be quite large, depending on the size of the original Grid object!), but will not result in any change to the data in the original HDF-EOS file.
- You will later need to use the IDL Tool (q.v.) to view or manipulate the output flat file.
- The flat file output option may be repeated as desired.

10 To end the session with displaying Grid object, in the GUI labeled **EOSView - Swath/Grid**, click on the **File** menu and select **Close**.

- The **EOSView - Swath/Grid** GUI will disappear.

26.8.5.1.4 Viewing HDF-EOS Swath Objects

This procedure describes how to use the EOSView tool to view science data in the HDF-EOS output file that are in HDF-EOS Swath format. These are generally the science data and not browse images.

The following is a list of tools and/or assumptions:

1. The SSIT Manager is running.
2. The output file is HDF-EOS and has been created and populated with metadata using the SDP Toolkit metadata tools.
3. At least one object is an HDF-EOS Swath.

Viewing HDF-EOS Swath Objects

1 From the SSIT Manager, select **Tools→ Product Examination → EOSView** from the menu.

- The EOSView GUI will be displayed.

2 In the GUI labeled **EOSView - *MyOutputFile.hdf***, double click on a Swath object listed for which data is to be inspected.

- A GUI labeled **EOSView - Swath Select** will be displayed.
- Information on **Dimensions, Geolocation Mappings, Indexed Mappings, Geolocation Fields**, **Attributes** for the selected Swath Object can be displayed by clicking on the corresponding checkboxes.

3 In the GUI labeled **EOSView - Swath Select**, click on the **Data Fields** checkbox and then click on the **OK** button. Then double click on one of the data fields listed.

- A GUI labeled **EOSView - Swath - *SwathObjectName* - Start/Stride/Edge** will be displayed where *SwathObjectName* will be replaced by the name of the Swath object selected in step 1.

4 To display the data in the form of a table of values, in the GUI labeled **EOSView - Swath - *SwathObjectName* - Start/Stride/Edge**, click on the checkboxes for both **ScanLineTra** and **PixelsXtrac** and then click on the **OK** button.

5 To display the data field in image form, in the GUI labeled ***MyDataField***, click on the **File** menu and then select **Make Image**.

- A GUI labeled **EOSView - Swath/Grid Image** will appear.

6 Optional colorization, zooming, panning while zooming features can be used in the GUI labeled **EOSView - Swath/Grid Image** to obtain your desired image.

- To end the session with displaying Swath object, in the GUI labeled **EOSView - Swath/Grid**, click on the **File** → **Close**.

26.8.5.1.5 Viewing HDF SDS Objects

This procedure describes how to use the EOSView tool to view science data in the HDF-EOS output file that are in HDF SDS (standard HDF science data set) format. To view an HDF SDS object with the EOSView tool, execute the procedure steps that follow:

The following is a list of tools and/or assumptions:

1. The SSIT Manager is running.
2. The output file is HDF-EOS and has been created and populated with metadata using the SDP Toolkit metadata tools.
3. At least one object is an HDF-SDS.

Viewing HDF SDS Objects

1 From the SSIT Manager, select **Tools**→ **Product Examination** → **EOSView** from the menu.

- The EOSView GUI will be displayed.

2 In the GUI labeled **EOSView - *MyOutputFile.hdf***, double click on a SDS object listed for which data is to be inspected.

- A GUI labeled **EOSView - Multi-Dimension SDS** will be displayed.

- A number of checkboxes will be displayed, one for each of the dimensions in the selected SDS (there will be at least two, an **X** and a **Y**).

3 In the GUI labeled **EOSView - Multi-Dimension SDS**, click on two of the dimension checkboxes and then click on the **Table** button. Then double click on one of the data fields listed.

- A GUI labeled ***MySDS*** will be displayed where ***MySDS*** will be replaced by the name of the SDS object selected in step 1.

4 To display the data field in image form, in the GUI labeled ***MySDS***, click on the **File** menu and then select **Make Image**.

- A GUI labeled EOSView - Image Display Window - ***MySDS*** will appear,

5 **Optional colorization, zooming, panning** while zooming can be used to obtain your desired output.

6 To end the session with displaying Swath object, in the GUI labeled **EOSView - Image Display Window - *MySDS***, select **File** → **Close** from the menu.

- The **EOSView - Image Display Window - *MySDS*** GUI will disappear.

26.8.5.2 Data Visualization with the IDL Tool

26.8.5.2.1 Viewing Product Data with the IDL Tool

The following procedures describe how to use the IDL (Interactive Data Language) COTS tool to inspect the data in the output file from a PGE. These procedures are geared toward binary and ASCII formats, but can be extended to other formats supported by IDL including HDF, NetCDF, and PGE. Consult the IDL references for details on these other formats.

The major activities addresses here include creating an image display, saving an image display, creating a plot display, and saving a plot display.

The following is a list of tools and/or assumptions:

1. The SSIT Manager is running.
2. The output file is binary, ASCII, or one of the other IDL supported data formats.
3. IDL has been properly installed and is accessible to the user.

Viewing Product Data with the IDL Tool

1 From the SSIT Manager, select **Tools** → **Product Examination** → **IDL** from the menu.

- An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 Select the procedure depending upon the activity to perform.

3 To end the IDL session, close any display windows remaining, then at the IDL prompt type **quit**, then press the **Enter** key.
- The IDL session will be closed.

26.8.5.2.2 Creating an Image Display Using IDL

The following procedure describes how to use the IDL Tool to create an image display.

The following is a list of tools and/or assumptions:

1. The SSIT Manager is running.
2. The PGE output file to be examined is of an IDL-supported type/format (if in doubt, consult the IDL Reference Guide.
3. IDL has been properly installed and is accessible to the user.
4. For binary files, data is assumed to be 8-bit characters.

Creating an Image Display Using the IDL Tool - Binary Data

1 From the SSIT Manager, select **Tools→ Product Examination → IDL** from the menu.
- An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 At the IDL prompt, type ***OPENR,1,'MyBinaryFilename'***, press the **Enter** key.
- The ***MyBinaryFilename*** is the full path name and file name of the binary data file of known dimensions to read in.
- The single quotes (') must be included around the path/file name.
- The **1** is the logical unit number.

3 At the IDL prompt, type ***MyImage=BYTARR(dimX, dimY)***, press the **Enter** key.
- The ***MyImage*** is the name to be given to the image once created.
- The ***dimX*** and ***dimY*** are the dimensions of the input data.

4 At the IDL prompt, type ***READU,1,MyImage***, press the **Enter** key.

5 At the IDL prompt, type ***TV,MyImage***, press the **Enter** key.
- The image, ***MyImage***, should then be displayed.
- Alternatively, type **TV,*MyImage,offsetx,offsety***, where offsetx and offsety are respectively the element and line offsets from MyImage's origin (because of the way IDL defines an image's origin, you may need to use negative values for these offsets).

6 At the IDL prompt, type ***LOADCT,3***, press the **Enter** key.
- This command loads color table number 3. Other color tables are available; refer to the *IDL Reference Guide* for more details.

7 At the IDL prompt, type ***CLOSE,1***, press the **Enter** key.

- This closes logical unit 1.
- Always close logical units or an error will result the next time an access is attempted.

Creating an Image Display Using the IDL Tool - ASCII Data

1 From the SSIT Manager, select **Tools→ Product Examination → IDL** from the menu.

- An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 At the IDL prompt, type ***OPENR,1,('MyASCIIfilename')***, then press the **Enter** key.

- The **MyASCIIfilename** is the full path name and file name of the ASCII data file of known dimensions to read in.
- The single quotes (') must be included around the path/file name.
- The **1** is the logical unit number.

3 At the IDL prompt, type ***MyImage=BYTARR(dimX,dimY)***, then press the **Enter** key.

- The ***MyImage*** is the name to be given to the image once created.
- The ***dimX*** and ***dimY*** are the dimensions of the input data.

4 At the IDL prompt, type ***READF,1,MyImage***, then press the **Enter** key.

5 At the IDL prompt, type ***TV,MyImage***, then press the **Enter** key.

- The image, **MyImage**, is displayed.
- Alternatively, type **TV,*MyImage,offsetx,offsety***, where offsetx and offsety are respectively the element and line offsets from MyImage's origin (because of the way IDL defines an image's origin, you may need to use negative values for these offsets).

6 At the IDL prompt, type ***LOADCT,3***, then press the **Enter** key.

- This command loads color table number 3. Other color tables are available; refer to the *IDL Reference Guide* for more details.

7 At the IDL prompt, type ***CLOSE,1***, then press the **Enter** key.

- This closes logical unit 1.
- Always close logical units or an error will result the next time an access is attempted.

Creating an Image Display Using the IDL Tool - JPEG Data:

1 At the IDL prompt, type **READ_JPEG,"*MyJPEGfilename.jpg*",*MyImage***, .

- The ***MyJPEGfilename*** is the full path name and file name of the JPEG formatted data file.
- The double quotes (") must be included around the path/file name.
- The ***MyImage*** is the name to be given to the image created.

2 At the IDL prompt, type **TVLCT,r,g,b**, .

- Note that r,g,b color table syntax is used for most formatted file types in IDL.

3 At the IDL prompt, type **TV,*MyImage***, .

- The image, ***MyImage***, should then be displayed.

26.8.5.2.3 Creating an Image Display Using the IDL Tool - PGM Data

1 From the SSIT Manager, select **Tools→ Product Examination → IDL** from the menu.

- An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 At the IDL prompt, type ***READ_PPM,"MyPGMfilename",MyImage,r,g,b***, then press the **Enter** key.

- The MyPGMfilename is the full path name and file name of the PGM formatted data file.
- The double quotes (") must be included around the path/file name.
- The MyImage is the name to be given to the image created.

3 At the IDL prompt, type ***TVLCT,r,g,b***, then press the **Enter** key.

- Note that r,g,b color table syntax is used for most formatted file types in IDL.

4 At the IDL prompt, type ***TV,MyImage***, then press the **Enter** key.

- The image, **MyImage**, should then be displayed.

26.8.5.2.4 Saving an Image Display Using IDL

The next procedure describes how to save an image display (once created) to either a data file or a graphic file.

The following is a list of tools and/or assumptions:

1. The SSIT Manager is running, IDL is running.
2. The PGE output file to be examined is of an IDL-supported type/format (if in doubt, consult the IDL Reference Guide.
3. For binary files, data is assumed to be 8-bit characters.

4. The image display is to be saved in a binary (8-bit) or ASCII (comma-delimited characters) format.

26.8.5.2.5 Save an Image Display Using IDL - Binary Data

1 From the SSIT Manager, select **Tools→ Product Examination → IDL** from the menu.

- An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 At the IDL prompt, type ***OPENW,1,('MyBinaryFilename.bin***'), then press the **Enter** key.

- The ***MyBinaryFilename.bin*** is the full path name and file name of the binary data file to write out.
- The single quotes (') must be included around the path/file name.
- The **1** is the logical unit number.

3 At the IDL prompt, type ***WRITEU,1,MyImage***, then press the **Enter** key.

- The ***MyImage*** is the name of the image to save.

4 At the IDL prompt, type ***CLOSE,1***, then press the **Enter** key.

- This closes logical unit 1.
- Always close logical units or an error will result the next time an access is attempted.

26.8.5.2.6 Save an Image Display Using IDL - ASCII Data

1 From the SSIT Manager, select **Tools→ Product Examination → IDL** from the menu.

- An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 At the IDL prompt, type ***OPENW,1,('MyASCIIfilename.asc')***, then press the **Enter** key.

- The ***MyASCIIfilename.asc*** is the full path name and file name of the binary data file to write out.
- The single quotes **(')** must be included around the path/file name.
- The **1** is the logical unit number.

3 At the IDL prompt, type ***PRINTF,1,MyImage***, then press the **Enter** key.

- The **MyImage** is the name of the image to save.

4 At the IDL prompt, type **CLOSE,1**, then press the **Enter** key.

- This closes logical unit 1.
- Always close logical units or an error will result the next time an access is attempted.

26.8.5.2.7 Save an Image Display Using JPEG Data

1 At the IDL prompt, type **WRITE_JPEG,*"MyJPEGfilename.jpg",MyImage***, press **Return**.

- The ***MyJPEGfilename.jpg*** is the full path name and file name of the JPEG data file to write out.

26.8.5.2.8 Creating a Plot Display Using IDL

The procedures for creating a plot display are clearly described in the IDL manuals; some exceptions are clarified below.

Setting axis limits for a plot:

1 At the IDL prompt, type **SURFACE,MyPlot,AX=70,AZ=70,xrange=[0,20],yrange=[0,20]zrange=[0,30]**, and press **Return**.

- The ***MyPlot*** is the IDL session variable (to which you have assigned some math function, program output, image, etc.).
- ***AX*** sets the displayed rotation about the X axis.
- ***AZ*** sets the displayed rotation about the Z axis.
- The values of ***xrange*** set the displayed portion of the X axis.
- The values of ***yrange*** set the displayed portion of the Y axis.
- The values of ***zrange*** set the displayed portion of the Z axis.
- The plot will then be displayed to the screen.

26.8.5.2.9 Setting Axis Titles for a Plot

1 At the IDL prompt, type **SURFACE,MyPlot,AX=70,AZ=70,xtitle='this is X', ytitle='this is Y',ztitle='this is Z'**, and press **Return**.

- The ***MyPlot*** is the IDL session variable (to which you have assigned some math function, program output, image, etc.).
- The value of ***xtitle*** sets the displayed title of the X axis.
- The value of ***ytitle*** sets the displayed title of the Y axis.
- The value of ***ztitle*** sets the displayed title of the Z axis.
- The plot will then be displayed to the screen.

26.8.5.2.10 Saving a Plot Display Using IDL

Saving a displayed plot to a permanent file:

1 At the IDL prompt, type **MyPlotDisplay=SURFACE,MyPlot,AX=80,AZ=20**, and press **Return**.

- The ***MyPlotDisplay*** is session name for the displayed plot of ***MyPlot***.
- The ***MyPlot*** is the IDL session variable (to which you have assigned some math function, program output, image, etc.).

2 At the IDL prompt, type **SAVE,MyPlotDisplay,4,'MyPlotOutput.ps'**, press **Return**.

- The ***MyPlotDisplay*** is the session name of the plot display .
- The ***MyPlotOutput.ps*** is the desired name for the saved file.
- The SAVE option number 4 sets the output file type to PostScript (ps). There are other options, of course (consult the IDL manuals).

26.8.5.3 Raster Math Fundamentals Using IDL

Most of this subject is covered in the IDL Manuals; some exceptions are described below.

Putting Raster Math results to a temporary display:

The following steps assume that you have already created two IDL session images, say "imageA" and "imageB", having the same raster size (same number of lines and elements).

1 At the IDL prompt, type **imageC=imageA+imageB-14.8**.

- The ***imageC*** is session name for the result of the Raster Math shown above. The operation indicated is performed on each pixel.
- IDL also supports other Raster Math operators—multiplication (*), division (/), exponent (^).

2 At the IDL prompt, type **TV, imageC**, .

- This displays the session image imageC, which can then be saved to a permanent file.

3 Alternatively (to steps 1 and 2), at the IDL prompt, type **TV,imageA*7.4/(imageB^8.2+1)**, .

- This shortcut displays the result of the indicated operation, but does not give you the option to either print the result from a session image or to save it as a permanent file. It may be useful (since your IDL session has a limited capacity for holding session images) if you are just experimenting with different Raster Math operations and do not intend to print or save your results.

26.8.5.4 Raster Mapping Fundamentals

This procedure describes how to use the IDL Tool to perform basic raster mapping functions. These are two-dimensional spatial functions involving map projections, rather than surface modeling (also called "2.5D"—for which see Plotting section and consult IDL manuals), volumetric modeling (also called "3D"—for which consult the IDL manuals), or two-dimensional spectral functions (for which see Raster Math section and consult the IDL manuals). Note that the pixel-level effects of Raster Mapping functions on an image are typically non-reversible; you can change an image's map projection from Projection A to Projection B and back again, but you wont get exactly the same image you started with (hint--you should make a back-up copy of your original image before engaging in Raster Mapping).

The following is a list of tools and/or assumptions:

1. The SSIT Manager is running.

2. IDL is running.

3. For a global data set image, one having coordinates defined from -180 to 180 degrees East Longitude and 90 to -90 degrees North Latitude, follow the steps listed under **Global Data Set Image**.

4. For a sub-global data set image, one having coordinates defined for subintervals of longitude and latitude (*e.g.* from -88 to -77 degrees East Longitude and 23 to 32 degrees North Latitude), follow the steps listed under **Sub-Global Data Set Image.**

26.8.5.4.1 Raster Mapping - Global Data Set Image

1 From the SSIT Manager, select **Tools→ Product Examination → IDL** from the menu.

- An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 At the IDL prompt, type ***TV,MyImage***, then press the **Enter** key.

- The ***MyImage*** is the image name of the global image data set.
- The image, ***MyImage***, should then be displayed.

3 At the IDL prompt, type ***MAP_SET,/ORTHOGRAPHIC***, then press the **Enter** key.

- IDL also supports other map projections. Refer to IDL Reference Guide.

4 At the IDL prompt, type ***MyNewImage=MAP_IMAGE(MyImage,startx,starty,/BILIN)***, then press the **Enter** key.

- The ***MyNewImage*** is the name to assign to the resulting image.
- The ***MyImage*** is the name of the original global image data set.

5 At the IDL prompt, type ***TV,MyNewImage,startx,starty***, then press the **Enter** key.

- The image ***MyNewImage*** should then be displayed.

6 Optional overlay Lat/Long. At the IDL prompt, type ***MAP_GRID***, then press the **Enter** key.

- This overlays Lat/Long graticule onto ***MyNewImage***.

7 Optional overlay world coastlines. At the IDL prompt, type ***MAP_CONTINENTS***, press the **Enter** key.

- This overlays world coastlines onto **MyNewImage**.
- **NOTE:** IDL also supports other resampling methods besides Bilinear Interpolation (***/BILIN***). Refer to *IDL Reference Guide*.

26.8.5.5 Raster Mapping - Sub-Global Data Set Image

For a sub-global data set image, one having geocentric-LLR coordinates defined for subintervals of longitude and latitude (e.g. from -88 to -77 degrees East Longitude and 23 to 32 degrees North Latitude.

The following is a list of tools and/or assumptions:

1. The SSIT Manager is running.
2. IDL is running.

1 From the SSIT Manager, select **Tools→ Product Examination → IDL** from the menu.

- An xterm (on the AIT Sun) will be displayed within which the IDL command interpreter will be run.

2 At the IDL prompt, type ***TV,MyImage***, then press the **Enter** key.

- The ***MyImage*** is the image name of the sub-global image data set.
- The image, ***MyImage***, should then be displayed.

3 At the IDL prompt, type ***MAP_SET,/MERCATOR,LIMIT=[lat1,lon1,lat2,lon2]***, then press the **Enter** key.

- Example: ***MAP_SET,/MERCATOR,LIMIT=[23,-88,32,-77],.***
- The ***lat1***, ***lon1***, ***lat2***, and ***lon2*** specify the latitude and longitude intervals of the sub-global image data set.

4 At the IDL prompt, type ***MyNewImage=MAP_IMAGE(MyImage,startx,starty,/BILIN.LATMIN=lat1,LATMAX=lat2,LONMIN=lon1,LONMAX=lon2)***, then press the **Enter** key.

- The ***MyNewImage*** is the name to assign to the resulting image.
- The ***MyImage*** is the name of the original global image data set.
- The ***lat1***, ***lon1***, ***lat2***, and ***lon2*** specify the latitude and longitude intervals of the sub-global image data set.

5 At the IDL prompt, type ***TV,MyNewImage,startx,starty***, then press the **Enter** key.

- The image ***MyNewImage*** should then be displayed.

6 Optional overlay Lat/Long. At the IDL prompt, type ***MAP_GRID***, then press the **Enter** key.

- This overlays Lat/Long graticule onto ***MyNewImage***.

7 Optional overlay world coastlines. At the IDL prompt, type ***MAP_CONTINENTS***, then press the **Enter** key.

- This overlays world coastlines onto ***MyNewImage***.
- Note that the IDL world coastline vector file is itself approximate; the match between this vector coastline and your image's own coastline will therefore also be approximate, but the potential mismatch will usually be too small to notice on a continental display. If your display is subcontinental, you may observe a noticeable mismatch between the vector and image coastlines—at this level of detail the approximate nature of the IDL vector file becomes noticeable, and the difference (if any) between underlying Earth Model (ellipsoid or horizontal datum) of the vector file and your map-projected image can add a substantial (up to 1+ km) additional mismatch to the display.

26.9 The ECS Assistant Functionality Replaced in Part by Scripts and Monitor GUI Whazzup

ECS (EOSDIS Core System) is a complex system comprising of many subsystems and components running on multiple heterogeneous host machines. The coordination of all the subsystems for SSI&T is an arduous, time consuming, error prone task. In order to improve our effectiveness and efficiency, an easy-to-use GUI tool, "ECS Assistant," has been developed to facilitate ECS SSI&T activities. Its origin as a development tool provides "fallback" functionality for other tools. Because of high overhead use of system resources, ECS Assistant has been scaled back in functionality.

Currently, the ECS Assistant has lost its capability to monitor ECS. This capability now exists with GUI Whazzup. ECS Assistant continues to have the ability to Monitor Subsystem functions and is mainly used for doing ECS Assistant Subsystem Installs (E.A.S.I.) and staging ESDT/DLL's into the directories, CUSTOM/data/ESS and CUSTOM/lib/ESS respectively. Database Review capability still exists however. The ESDT Manager Installation/Deletion functions are no longer available, the ability to startup and shutdown subsystem servers and the use of the ECS Logfile Viewer have been taken away.

The use of scripts provides users with the means to perform functions such as subsystem server startup and shutdown. During the course of performing these tasks, SSI&T operators can use the following Scripts to perform the following functions:

- To **start up or shut down all servers** at the same time, a Script is used that accesses a list of subsystem servers.
- To **start up and shut down servers individually** using a Script established within each subsystem.
- To **graphically monitor the server up/down status** with the **Whazzup GUI**
- To **view** ESDTs for SSI&T
- To **review various databases** used in the ECS system by using **an ISQL Browsers** established in each subsystem.

In the following sections, we will address aspects of how to use **Scripts, the Whazzup GUI** and **portions of ECS Assistant** in our SSI&T activities.

- **Section one** explains how to use **Scripts** to facilitate and manage the subsystems and their servers, including **server start up and shut down.**
- **Section two** contains an ESDT monitor function which includes reviewing the Science Data Server database through the **DB Viewer** GUI provided by ECS Assistant

26.9.1 Using Scripts to Start Up/Shut Down Servers

The DAACs may have established their own scripts to Start Up/Shut Down Subsystem servers. This procedure describes Scripts that are used at Landover on the VATC and PVC systems to start up and shut down subsystem servers. The procedure described here will apply to all the servers from different subsystems as well as individual servers.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow

26.9.1.1 All Servers Start Up/Shut Down

1 Script name: **rcmd_start_mode TS1, rcmd_kill_mode TS1**

2 Location: **/home/cmshared/bin/**

3 **HOSTLST** contains all the Subsystem Servers that will be operated upon.

26.9.1.2 Individual Servers Start Up/Shut Down

1 If you want to Start or Kill an individual server, you must be on the individual machine that supports the server in question.

2 To kill a server, identify its process ID by typing **ps –ef | grep <*xxx . . .*>**, where **<*xxx . . .*>** is the first part of the name of the server to be killed, and pressing the **Return/Enter** key.
- UNIX returns a list of process ID numbers for running processes that begin with the typed first part of the name.
- Note the process ID for the server to be killed, ensuring that it is the one in the mode where you wish to kill the server.

3 Type **kill -15 <*pid*>**, where **<*pid*>** is the process ID of the server process noted in step 2, and then press the **Return/Enter** key.
- The server should be killed; you can check the success of this step by repeating the command used in step 2, noting that the server process and its process ID are not returned.

4 To start an individual server, first change the working directory to the utilities directory by typing **cd /usr/ecs/<*MODE*>/CUSTOM/utilities** and then pressing the **Return/Enter** key.

- The working directory is changed to **/usr/ecs/<*MODE*>/CUSTOM/utilities**.

5 Next, locate the name of the start script for the server to be started by typing **ls -al** and then pressing the **Return/Enter** key.

- UNIX returns a list of the utilities in the directory.

6 Type ***<startscriptname> <MODE>***, where ***<startscriptname>*** is the name of the start script for the server to be started and ***<MODE>*** is the mode in which the server is to be started.

- The server should be started; you can check the success of this step by repeating the command used in step 2, noting that the server process is returned with a process ID to indicate that it is running.

26.9.2 Bringing Up ECS Assistant

Assumptions:

1. The ECS Assistant has been properly installed.
2. The required environment variables have been set properly.

NOTE: SGI machines are not a part of ECS Assistant functionality.

26.9.2.1 Run the ECS Assistant

1 At the UNIX Console or Terminal **setenv DISPLAY clientname:0.0** and then press the **Enter** key.

2 Create an xterm by typing: xterm -n hostname &

- The ***hostname*** is the name of the machine on which the ECS Assistant is to be displayed, *i.e.*, the machine that you are using.

3 Log into one of the host machines used for SSIT, (Tested using **telnet p0acs03** for **SDSRV** and **p0ins02** for **ID:, PW:**

4 At the UNIX Console or Terminal type **setenv DISPLAY clientname:0.0** and then press the **Enter** key**.**

- To verify the setting, type **echo $DISPLAY** and then press the **Return/Enter** key.
- **setenv ECS_HOME/usr/ecs**
- **setenv TK_LIBRARY/tools/lib/tk4.2**
- (Mount point called /tools must be mounted.)

5 If necessary, at the UNIX prompt on the host from which the ECS Assistant is to be run, type **cleartool setview *ViewName*** and then press the **Enter** key.

- The ***ViewName*** is the ClearCase view to be used while the ECS Assistant is running in this session. For example, type: cleartool jdoe and then press the **Enter** key.
- A ClearCase view is required only if the ECS Assistant needs to be able to "see" into a ClearCase VOB; a view is not necessary otherwise.

6 At the UNIX prompt, type **EA** and then press the **Enter** key.

- File **/tools/common/ea** must exist in the path. (This can be set in the **.cshrc** or **.kshrc** file)
- EA is an alias for: /tools/common/ea is the path where ECS Assistant is installed.
- This will invoke the ECS Assistant GUI with three push buttons for selecting the proper activities.

7 At the ECS Assistant GUI, click the **Subsystem Manager** pushbutton.

- This will invoke the Subsystem Manager GUI.

8 Select a mode by clicking a mode in the mode listing. The mode should be the one to be used for SSI&T.

- Once the mode is selected, the color of the subsystem name list is changed.

9 Select a subsystem with the **Subsystem** radio button.

- The component list for the selected subsystem will appear in the component window.

10 Select a component by clicking the component name under the component window.

- The selected component will be highlighted.
- The server list corresponding to that component will appear in the server window.

11 Select a server by clicking the server name from the server list under the servers window.

- The server selected is highlighted.
- Refer to the section titled: "Using Scripts to Start Up / Shut Down Servers."

12 To exit the Subsystem Manager GUI, select **File..Exit** in the menu bar of the Subsystem Manager GUI.

- This will terminate the Subsystem Manager GUI.

26.9.3 Monitoring ECS Using the WHAZZUP Web GUI

To use the WHAZZUP GUI to monitor ECS make the following entries to display the GUI.

1 Log into a server that has Web access.

2 **setenv DISPLAY ..:0.0**

3 **/tools/bin/ssh -l cmshared t1code1u.ecs.nasa.gov**

4 Password:

5 **setenv DISPLAY :0.0**

6 **ssh t1pls02**

7 **/tools/bin/netscape/** when the Netscape GUI is displayed, then select from the Bookmark: **whazzup.** A GUI should appear on the terminal screen.
- Other links listed in the bookmark List go to sites that support ECS.

26.9.4 Using ECS Assistant to View the ECS Science Data Server Database

ESDTs and their granules can be viewed in the ECS Science Data Server database. ECS Assistant provides an easy way to review the records stored in this database by using the ECS Assistant DB Viewer. There are two main windows in the DB Viewer. The first is called Collections and is used to display ESDT information included in the Collection database table. Information listed in this table includes ESDT short names, times last updated, types, etc. If an ESDT is added to the Science Data Server, its record will be shown in this window. The other window is called Granules and is used to display information included in the Granule database table. If a granule is inserted for an ESDT, the granule information will be listed in this window if its ESDT is highlighted in the Collection window. In addition to these two main windows, this DB Viewer GUI can also show ESDT database validation rules, PSA information, and summary information about the database reviewed.

Detailed procedures for the SSI&T operator to perform are provided in the sections that follow.

Assumptions:

1. The ECS Assistant has been properly installed.
2. The ECS Subsystem Manager is running.
3. The environment variables for using the database have been set correctly.

To start up the ECS monitor GUI, execute the procedure steps that follow:

1 Follow Section Bringing Up ECS Assistant to invoke the ECS Assistant GUI.
- The ECS Assistant GUI will be launched.

2 At the ECS Assistant GUI, select ESDT Manager GUI by clicking the ESDT Manager.
- The ESDT manager GUI will appear.

3 At the ECS ESDT Manager GUI, select the DB Viewer by clicking the **DB Viewer** button.
- The Database Login GUI will appear.

- Fill in the fields to point to the specific database for the mode used.
- Click Login to open the DB Viewer.
- The DB Viewer GUI will appear.
- ESDTs are listed in the Collections window.

4 To view the inserted granules for a selected ESDT, first select an ESDT by clicking its short name in the Collections window.
- The selected ESDT is highlighted.
- Granule information for that ESDT, if there is any, will be listed in the Granules window.

5 To exit, click the **EXIT** button.
- This will end the DB Viewer GUI.

Using ECS Assistant to View ECS Science Data Server Database

1 Invoke the ECS Assistant GUI.

2 At the ECS Assistant GUI, invoke the **ESDT Manager**.

3 At the ECS ESDT Manager GUI, select the **DB Viewer** to open the login window for the database.

4 Fill in the Login field and open **DB Viewer** for the selected database.

5 View the inserted granules for a selected ESDT.

26.10 ESDT Management

In order for science data to be handled by ECS, it must be formerly described. At the collection level, that description is the Earth Science Data Type or ESDT. Basically, when an ESDT is defined/installed to the data server subsystem, the SDSRV parses the descriptor into various portions needed by its own CSCIs, and other subsystems.

An entry is made in the SDSRV database containing the meaning of the ESDT, each of its attributes, and each of its services (references to the executable DLLs). The (Sybase) database managed by the SDSRV has sufficient information to satisfy queries sent to the SDSRV.

The ESDT Descriptor file text contains the information mentioned above in an ODL format. The bulk of these files are placed in a given mode during the ECS install process for that mode. They generally reside in directory /usr/ecs/<*MODE*>/CUSTOM/data/ESS. In order for these descriptors to be of any use, their information needs to be extracted and parsed to various subsystem databases. This is called the ESDT Install Process.

Also, the ESDT Descriptor files may contain errors or the basic ESDT information is evolved such that the old descriptor information may have to be replaced or updated in the relevant databases.

This section will describe how to Check, Install, Remove or Update an ESDT.

26.10.1 Inspecting ESDTs

Before installing or updating an ESDT, one needs to check for its existence. Also, one may want to examine the contents of an ESDT, e.g., what does the header say about the latest changes and when they were made. These types of checks/inspections can be performed with general UNIX tools or with the Science Data Server GUI.

Assumptions:

1. The required environment variables (e.g., DISPLAY) have been set properly.
2. The Science Data Server (EcDsScienceDataServer) for the desired mode is running.
3. The sybase server for the Science Data Server database (e.g., for PVC: p0acg05_srvr) is running.

To bring up the Science Data Server GUI, execute the steps that follow:

1 Log into the Science Data Server host. The file .sitemap under /usr/ecs/<*MODE*>/CUSTOM contains this information. Following established ECS DAAC host naming conventions, some examples are: for the PVC - p0acs03, for the GDAAC - g0acs3.

2 At the UNIX prompt on the host from which the Science Data Server GUI is to be run, type **cd /usr/ecs/<*MODE*>/CUSTOM/utilities .**

3 Next, type **EcDsSdSrvGuiStart <*MODE*>**, press **Return**.
- The Science Data Server GUI will be presented.

4 Scroll thought the Data Type Information box. If the shortname and version of the ESDT is displayed, then that ESDT was installed at least to the Science Data Server database.
- For an ESDT to be fully installed and useful to ECS, there are two other databases that need to have been successfully affected by the installation process. These are the Data Dictionary and Subscription databases.
- If the shortname/version of the desired ESDT does not appear in the box as mentioned, it can be assumed it is not installed in the Science Data Server database. However, it could have been installed in one or more of the other three databases.

5 To view the contents of an ESDT Descriptor File, select the shortname/version in the box and click on View.

- Alternately, one can use a text editor and open up the corresponding ESDT Descriptor file. The installed version of the Descriptor file resides in /usr/ecs/<MODE>/COMMON/cfg/DsESDTDesc. During the ESDT installation process, the descriptor file is copied from /usr/ecs/<MODE>/COMMON/data/ESS into /usr/ecs/<MODE>/COMMON/cfg/DsESDTDesc.
- The descriptor file contains version/date information in its header. The rest of the file is in ODL metadata format and describes the corresponding ESDT to the ECS.

Table 26.10.1-1. ESDT Inspection Using Science Data Server GUI - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	Logon to Science Data Server platform	Log onto SDSRV host
2	type **cd /usr/ecs/<*MODE*>/CUSTOM/utilities**	Press **Return**
3	type **EcDsSdSrvGuiStart <*MODE*>**	Press **Return**
4	Scroll through Data Type Information box	none
5	Select shortname/version	Click the **View** button

26.10.2 Removing ESDTs

If one wants to install a modified version of an ESDT, retaining the shortname and version number, there are two ways to go. The first way is to completely replace the descriptor file and corresponding ESDT data in the four relevant databases. Then, perform an ESDT add. The second way, if the nature of the old and new ESDT permits, is to perform an Update. The advantages/disadvantages are summarized below:

ESDT ADD Advantages	**Disadvantages**
Can be done with any ESDT	Prior to Add, ESDT removal scripts must be run for all affected databases. All granule pointer information is lost for granules belonging to that ESDT.

ESDT Update Advantages	**Disadvantages**
No removal scripts need to be run Granule references are preserved.	Only certain ODL structures supported. Science Data Server must be brought up with a StartTemperature=maintenance. This means it is unusable by anything else until it is recycled with an operational StartTemperature. Many other conditions must be met.

26.10.2.1 ESDT Removal Using ContributionDriverStart

Assumptions:

1. The required environment variables (e.g., **DISPLAY**) have been set properly.
2. The sybase servers for the three involved databases are working properly. e.g., for the PVC:

   p0acg05_srvr ---- supports the (DSS) Science Data Server databases
   p0ins01_srvr ---- supports the (IDG/CSS) Subscription Server databases
   p0ins02_srvr ---- supports the (DMS) Data Dictionary databases

3. The user knows the various ids, passwords and parameters required by the four scripts that will be used.

To remove an ESDT from the ECS for a given mode, execute the steps that follow:

Removal From the Data Dictionary Database

1 Log into the (DMS) Data Dictionary Server host. The file .sitemap under /usr/ecs/<*MODE*>/CUSTOM contains this information. Following established ECS DAAC host naming conventions, some examples are: for the PVC - p0ins02, for the GDAAC - g0ins02.

2 At the UNIX prompt on said host , type **cd /usr/ecs/<*MODE*>/CUSTOM/dbms/DMS**, press **Return** .

- This subdirectory contains the script to be run - **DmDbCleanCollection**.

3 Prior to running the script, four environmental parameters need to be assigned proper values. These parameters are DSQUERY, DBNAME, DBUSERNAME, and DBPASSWD. DSQUERY points to the sybase server that contains the Data Dictionary database. DBNAME is the name of the Data Dictionary database, DBUSERNAME is the Data Dictionary login ID. DBPASSWD is the Data Dictionary login password. To set these environmental parameters in C shell, follow the sample steps that follow. Values are for the PVC in mode TS1. DBUSERNAME and DBPASSWD values won't be displayed here for security reasons.

- At the UNIX prompt ,
  type **setenv DSQUERY p0ins02_srvr**, press **Return** .
- At the UNIX prompt ,
  type **setenv DBNAME EcDmDictService_TS1**, press **Return** .
- At the UNIX prompt,
  type **setenv DBUSERNAME ************, press **Return** .
- At the UNIX prompt,
  type **setenv DBPASSWD *********, press **Return** .

4 The script to be run requires two arguments - an ESDT shortname and the version number for said shortname. The form is **DmDbCleanCollection *ShortName Version*** . A PVC example would be:

- At the UNIX prompt, type **DmDbCleanCollection AE_5Dsno 001,** press **Return**

Removal From the Subscription Database

1 Log into the (CSS/IDG) Subscription Server host. The file .sitemap under /usr/ecs/<*MODE*>/CUSTOM contains this information. Following established ECS DAAC host naming conventions, some examples are: for the PVC - p0ins01, for the GDAAC - g0ins01.

2 At the UNIX prompt on said host , type **cd /usr/ecs/<*MODE*>/CUSTOM/utilities**, press **Return** .

- This subdirectory contains the script to be run - **dbDeleteEvents.csh**.

3 This script requires five arguments. This will be of the form **dbDeleteEvents.csh *mode ShortName Version ID UserName PassWd* .**

- The arguments mean:
  mode The DAAC mode to be affected by the script
  ShortName The ShortName of the ESDT to be removed
  VersionID The version of ShortName to be removed
  UserName The Subscription database login username
  PassWd The Subscription database login password
- For example, to remove the ESDT with ShortName **AE_5Dsno** (version 1) from the Subscription database in mode TS1 on the PVC, at the UNIX prompt , type **dbDeleteEvents.csh TS1 AE_5Dsno 001 ******* *******,**
  press **Return**.
  Note: for security reasons, the actual database login ID and Password are not shown.

Removal From the Science Data Server Database

1 Log into the (DSS/SDSRV) Science Data Server host. The file .sitemap under /usr/ecs/<MODE>/CUSTOM contains this information. Following established ECS DAAC host naming conventions, some examples are: for the PVC - p0acs03, for the GDAAC - g0acs03.

2 At the UNIX prompt on said host , type **cd /usr/ecs/<*MODE*>/CUSTOM/utilities**, press **Return** .

- This subdirectory contains the script to be run - **EcDsSrRmesdt** .

3 This script requires at least two arguments. This will be of the form **EcDsSrRmesdt *mode DescripFileName1 DescripFileName2 …. .***

- The arguments mean:
  mode The DAAC mode to be affected by the script

DescripFileName1 The name of an ESDT descriptor file that corresponds to the ESDT to be removed.

DescripFileNameN More than one ESDT may be removed when running the script **EcDsSrRmesdt.** For each ESDT to be removed, a corresponding ESDT descriptor file name is required.

- For example, to remove the ESDT with ShortName **AE_5Dsno** (version 1) from the Science Data Server database in mode TS1 on the PVC, at the UNIX prompt ,
  type **EcDsSrRmesdt TS1 DsESDTAmAE_5Dsno.001.desc**,
  press **Return**.
  Note01: The ESDT descriptor files are located in two subdirectories. The "holding area" is /usr/ecs/<mode>/CUSTOM/data/ESS . They are first put here during mode drop installations or manually as new versions are delivered. When an ESDT is installed into the Science Data Server database, the descriptor file is copied from the "holding area" and placed into /usr/ecs/<mode>/CUSTOM/cfg/DsESDTDesc .
  Note02: The ESDT version ID number is incorporated in the descriptor file name as exemplified by the location of ".001" in **DsESDTAmAE_5Dsno.001.desc** .

Table 26.10.2.1-1. ESDT Removal Using ContributionDriverStart - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
	Removal From Data Dictionary Server DB	
1	Logon to DMS platform	Log onto DMS host
2	type **cd /usr/ecs/<*MODE*>/CUSTOM/dbms/DMS**	Press **Return**
3a	type **setenv DSQUERY *sybase_ddict_srvr***	Press **Return**
3b	type **setenv DBNAME *EcDmDictService_dbname***	Press **Return**
3c	type **setenv DBUSERNAME ***********	Press **Return**
3d	type **setenv DBPASSWD ***********	Press **Return**
4	type **DmDbCleanCollection *ShortName VersionID***	Press **Return**
	Removal From Subscription Server DB	
1	Logon to (CSS/IDG) Subscription Server platform	Log onto CSS/IDG host
2	type **cd /usr/ecs/<*MODE*>/CUSTOM/utilities**	Press **Return**
3	type **dbDeleteEvents.csh *mode ShortName VersionID UserName Passwd***	Press **Return**
	Removal From Science Data Server DB	
1	Logon to (DSS) Science Data Server platform	Log onto DSS/SDSRV host
2	type **cd /usr/ecs/<*MODE*>/CUSTOM/utilities**	Press **Return**
3	type **EcDsSrRmesdt mode DescriptFileName1 …**	Press **Return**

26.10.3 Adding ESDTs

Generally, an ESDT or group of ESDTs are Added using the Science Data Server GUI. The single biggest disadvantage of doing an ESDT add over performing an ESDT update is that one must remove the ESDT (if it exists) before performing the add. Aside from the labor involved removing an ESDT, once the ESDT is removed, all granule pointers in the various databases for that ESDT are lost. If one wants to reference the granules after the revised ESDT is added, the granules will have to be reinserted. This would definitely impact ongoing operations.

The main advantage of adding an ESDT is that the Add operation itself is fairly simple.

26.10.3.1 Adding an ESDT Using the Science Data Server GUI

Assumptions:

1. The required environment variables (e.g., **DISPLAY**) have been set properly.

2. The sybase servers for the three involved databases are working properly. e.g., for the PVC:

   p0acg05_srvr ---- supports the (DSS) Science Data Server databases
   p0ins01_srvr ---- supports the (IDG/CSS) Subscription Server databases
   p0ins02_srvr ---- supports the (DMS) Data Dictionary databases

3. The subsystem servers associated with the three involved databases are running. e.g., for the PVC, mode TS1:

Platform	**Server**
p0acs03	/usr/ecs/TS1/CUSTOM/bin/DSS/EcDsScienceDataServer
p0ins01	/usr/ecs/TS1/CUSTOM/bin/CSS/EcSbSubServer
p0ins02	/usr/ecs/TS1/CUSTOM/bin/DMS/EcDmDictServer

5. If the ESDT to be added already exists, it will be removed from the involved databases before the Add operation takes place.

To Add an ESDT to the ECS for a given mode, execute the steps that follow:

1 Log into the (DSS) Science Data Server host. The file .sitemap under /usr/ecs/<MODE>/CUSTOM contains this information. Following established ECS DAAC host naming conventions, some examples are: for the PVC - p0acs03, for the GDAAC - g0acs03.

2 At the UNIX prompt on said host , type **cd /usr/ecs/<*MODE*>/CUSTOM/utilities**, press **Return** .

3 Type **EcDsSdSrvGuiStart *mode***, press **Return**
- This brings up the Science Operator Data Server GUI.

4 Click the "Add…" button on the Science Data Server Operator GUI.
- This brings up the "Add Data Type" GUI.

5 The user can type in the descriptor file name for the ESDT to be added. However, it is usually easier and less prone to error, to click the "File…" button.

- This brings up the "Add ESDT" GUI.

6 The "Add ESDT" GUI displays a list of the eligible descriptor files that can be used to add an ESDT. One can tailor this list by making the appropriate entry in the "Filter" box.

- Select one or more descriptor file names from the file list. To select more than one file name, the user needs to hold down the Control or Shift key while clicking on the desired file names.
- For each descriptor file selected, the corresponding ESDT will be added when the process is complete.

7 When the desired descriptor files have been selected, click in the "OK" box of the "Add ESDT" GUI.

- The "Add Data Type" GUI will become populated with the descriptor file selections.

8 Click the "OK" box in the "Add Data Type" GUI.

- The Add ESDT process will start.
- The "Operator Messages" will display the status of the install process.
- If the install process seems to have errors, go to the /usr/ecs/<mode>/CUSTOM/logs directories on the various server platforms and check the logs for the servers involved as well as for the "Science Data Server Operator GUI" log.
- If everything went well, the selected ESDTs have been added.

Table 26.10.3.1-1. Adding ESDTs With the Science Data Server GUI

Step	What to Enter or Select	Action to Take
1	Logon to Science Data Server platform	Log onto SDSRV host
2	type **cd /usr/ecs/<*MODE*>/CUSTOM/utilities**	Press **Return**
3	type **EcDsSdSrvGuiStart <*MODE*>**	Press **Return**
	SDSRV GUI appears	none
4	Click the "Add…" button in the SDSRV GUI	none
5	Click the "File…" button in the "Add Data Type" GUI	none
6	Select descriptor file names in "Add ESDT" GUI	none
7	Click **OK** in the "Add ESDT" GUI	none
8	Click **OK** in the "Add Data Type" GUI	none

26.10.4 Updating ESDTs

Generally, an ESDT or group of ESDTs are Updated using the Science Data Server GUI. There are certain conditions that a given set of descriptors must meet in order for an Update to be possible. Furthermore, the Science Data Server has to be running with a StartTemperature value of **"maintenance"** in order for the Update function to work. This means the mode involved is unusable by anyone else. The main advantage of performing an Update is that the old ESDT doesn't have to be removed first, thus preserving the ECS's knowledge of any granules that were inserted with the ESDT shortname that is being Updated. This is definitely a plus for an operational mode.

Following is a listing of what the Update ESDT Capability can do. Implicit in this are the things it can't do.

- Add optional Collection level metadata
- Add optional Inventory level metadata (including Product Specific Attributes (PSAs))
- Add additional services
- Add additional events
- Add new parameters to existing services
- Add qualifiers to existing events
- Add additional valid values to Inventory level metadata attributes
- Change values of single and multi-value Collection level metadata
- attributes (exceptions: AdditionalAttributeName, AdditionalAttributeType,
- AnalysisShortName, CampaignShortName, InstrumentShortName,
- PlatformShortName, SensorCharacteristicName, SensorCharacteristicType,
- SensorShortName, ShortName, VersionID, and type)
- Change a mandatory attribute to optional
- Modify parameters in existing services

One thing an ESDT Update explicitly can't do:

- Add Mandatory attributes.

It is clear from the above, that before an ESDT Update is attempted, consultation with an ESDT specialist is advised.

26.10.4.1 Updating an ESDT Using the Science Data Server GUI

Assumptions:

1. The required environment variables (e.g., **DISPLAY**) have been set properly.
2. The sybase servers for the three involved databases are working properly. e.g., for the PVC:

   p0acg05_srvr ---- supports the (DSS) Science Data Server databases
   p0ins01_srvr ---- supports the (IDG/CSS) Subscription Server databases
   p0ins02_srvr ---- supports the (DMS) Data Dictionary databases

3. The subsystem servers associated with the three involved databases are running. e.g., for the PVC, mode TS1:

Platform	Server
p0acs03	/usr/ecs/TS1/CUSTOM/bin/DSS/EcDsScienceDataServer
p0ins01	/usr/ecs/TS1/CUSTOM/bin/CSS/EcSbSubServer
p0ins02	/usr/ecs/TS1/CUSTOM/bin/DMS/EcDmDictServer

To Update an ESDT to the ECS for a given mode, execute the steps that follow:

1 Log into the (DSS) Science Data Server host. The file .sitemap under /usr/ecs/<MODE>/CUSTOM contains this information. Following established ECS DAAC host naming conventions, some examples are: for the PVC - p0acs03, for the GDAAC - g0acs03.

2 At the UNIX prompt on said host , type **cd /usr/ecs/<*MODE*>/CUSTOM/utilities**, press **Return** .

3 Type **EcDsSdSrvGuiStart *mode***, press **Return**

- This brings up the Science Operator Data Server GUI.

4 Click the "Update…" button on the Science Data Server Operator GUI.

- This brings up the small "Update ESDT" GUI.

5 The user can type in the descriptor file name for the ESDT to be Updated. However, it is usually easier and less prone to error, to click the "File…" button.

- This brings up the large "Update ESDT" GUI.

6 The large "Update ESDT" GUI displays a list of the eligible descriptor files that can be used to update an ESDT. One can tailor this list by making the appropriate entry in the "Filter" box.

- Select one or more descriptor file names from the file list. To select more than one file name, the user needs to hold down the Control or Shift key while clicking on the desired file names.
- For each descriptor file selected, the corresponding ESDT will be Updated when the process is complete.

7 When the desired descriptor files have been selected, click in the "OK" box of the large "Update ESDT" GUI.

- The small "Update ESDT" GUI will become populated with the descriptor file selections.

8 Click the "OK" box in the small "Update ESDT" GUI.

- The Update ESDT process will start.
- The "Operator Messages" will display the status of the install process.
- If the install process seems to have errors, go to the /usr/ecs/<mode>/CUSTOM/logs directories on the various server platforms and check the logs for the servers involved as well as for the "Science Data Server Operator GUI" log.
- If everything went well, the selected ESDTs have been Updated.

Table 26.10.4.1-1. Updating ESDTs With the Science Data Server GUI

Step	What to Enter or Select	Action to Take
1	Logon to Science Data Server platform	Log onto SDSRV host
2	type **cd /usr/ecs/<*MODE*>/CUSTOM/utilities**	Press **Return**
3	type **EcDsSdSrvGuiStart <*MODE*>**	Press **Return**
	SDSRV GUI appears	none
4	Click the "Update…" button in the SDSRV GUI	none
5	Click the "File…" button in the large "Update ESDT" GUI	none
6	Select descriptor file names in large "Update ESDT" GUI	none
7	Click **OK** in the large "Update ESDT" GUI	none
8	Click **OK** in the small "Update ESDT " GUI	none

26.10.5 ESDT Volume Group Configuration

Once an ESDT is installed into the ECS, the system knows how to deal with the collections and granules associated with that ESDT - up to a point. The Storage Management subsystem needs some additional information for its database so that it knows where to archive and retrieve the data associated with a given ESDT. This is the ESDT Volume Group information. When an Insert or Acquire is performed, Storage Management needs to know which HWCI (Hardware CI) and directory are involved.

This Volume Group information can be created and modified using the Storage Management GUI. The GUI start script is EcDsStmgtGuiStart and resides in the standard utilities directory for each mode. This GUI normally resides on the (DSS) Data Distribution Server platform. In the PVC, for example, that platform is p0dis02 .

For the official baseline document that describes ESDTs and includes the configured Volume Group information, access http://pete.hitc.com/baseline/index.html , click on Technical Documents and then pick the row that looks like "019 ESDT Baseline 910-TDA-019-Rev*xx*.xls"

26.10.5.1 Modifying an ESDT's Volume Group Information

Assumptions:

1. The required environment variables (e.g., **DISPLAY**) have been set properly.

2. The sybase server for the Storage Management Subsystem databases is working properly. e.g., for the PVC, p0acg05_srvr.

To modify a given ESDT's Volume Group information for a given mode, execute the steps that follow:

1 Log into the (DSS) Data Distribution Server host. The file .sitemap under /usr/ecs/<MODE>/CUSTOM contains this information. Following established ECS DAAC host naming conventions, some examples are: for the PVC - p0dis02, for the GDAAC - g0dis02.

2 At the UNIX prompt on said host , type **cd /usr/ecs/<*MODE*>/CUSTOM/utilities**, press **Return** .

3 Type **EcDsStmgtGuiStart *mode***, press **Return**

- This brings up the Storage Management Control GUI.

4 Click the "Vol Grp Config" tab on the GUI

- This brings up the "Volume Group Information" pane.

5 Select an ESDT to be modified by scrolling through the Volume Group Information pane and clicking on the ShortName.VersionID desired.

6 Click the Modify button.

- The Modify Volume Groups GUI will appear

7 Type in only the information that needs to be changed.

- The HWCI information must be selected from a list that is brought up by clicking on the New HWCI selection arrow.

8. Click the "OK" box when satisfied with the modifications entered.

- The Update ESDT process will start.
- The "Operator Messages" will display the status of the install process.

Table 26.10.5.1-1. Changing ESDT Volume Group Information

Step	What to Enter or Select	Action to Take
1	Logon to (DSS) Data Distribution Server host	Log onto DDIST host
2	type **cd /usr/ecs/<*MODE*>/CUSTOM/utilities**	Press **Return**
3	type **EcDsStmgtGuiStart <*MODE*>**	Press **Return**
	Stmgt Control GUI appears	none
4	Click the "**Vol Grp Config"** tab	none
5	Click on the ***ShortName.VersionID*** to be modified	none
6	Click the **Modify** button	none
	The Modify Volume Groups GUI will appear	
7	Type in the new information	none
8	Click **OK** button	none

26.10.5.2 Adding an ESDT's Volume Group Information

Assumptions:

1. The required environment variables (e.g., **DISPLAY**) have been set properly.
2. The sybase server for the Storage Management Subsystem databases is working properly; e.g., for the PVC, p0acg05_srvr.

To enter information for an ESDT not already entered into the Storage Management's database (i.e., the ESDT.VersionID doesn't appear in the Volume Group Information pane of the Storage Management Control GUI) for a given mode, execute the steps that follow:

1 Log into the (DSS) Data Distribution Server host. The file .sitemap under /usr/ecs/<MODE>/CUSTOM contains this information. Following established ECS DAAC host naming conventions, some examples are: for the PVC - p0dis02, for the GDAAC - g0dis02.

2 At the UNIX prompt on said host , type **cd /usr/ecs/<*MODE*>/CUSTOM/utilities**, press **Return** .

3 Type **EcDsStmgtGuiStart *mode***, press **Return**

- This brings up the Storage Management Control GUI.

4 Click the "Vol Grp Config" tab on the GUI

- This brings up the "Volume Group Information" pane.

5 Click the "Add…" button.

- The Add Volume Group GUI will appear.

6 Type in all the required information.

- The HWCI information must be selected from a list that is brought up by clicking on the HWCI selection arrow.

7 Click the "OK" box when satisfied with the information entered.

- The Add Volume Group process will start.
- The "Operator Messages" will display the status of the install process.

Table 26.10.5.2-1. Adding an ESDT's Volume Group Information

Step	What to Enter or Select	Action to Take
1	Logon to (DSS) Data Distribution Server host	Log onto DDIST host
2	type **cd /usr/ecs/<*MODE*>/CUSTOM/utilities**	Press **Return**
3	type **EcDsStmgtGuiStart <*MODE*>**	Press **Return**
4	Click the "**Vol Grp Config"** tab	none
5	Click the "**Add…"** button	none
6	Enter all the requisite information	none

26.11 Production Planning Considerations

During normal operations it is expected that the Production Planner will not have to add PRs to the PDPS database very frequently. The frequency of this activity is, to some extent, determined by the SCF responsible for the science software.

- The PR is a template request to generate a particular data product and results in a production run of the associated SCF-provided PGE.
- PR specifies a range (temporal, orbit, or tile) over which the data products are to be produced or the PGEs are to be scheduled.
- PR might request that the data product be produced for only a single day's data.
- PR might request that data products be produced for every opportunity of input data for several months, resulting in several hundred jobs being planned and run as the input data become available.
- Early in a mission the SCF may prefer to request processing for a short time period only (e.g., a week or less).
- At that time the SCF is gaining an understanding of the on-orbit behavior of the instrument, the resulting data, and the interaction of the science processing software with real data.

- SCF reviews the quality of the products and notifies the Production Planner of the need for any changes to the PR (e.g., discontinue the PR, change time ranges, or modify input parameters).
- When the SCF has developed a good understanding of the instrument's behavior, the team may be comfortable requesting processing for months at a time.
- DAAC operations may have operational reasons for wanting to issue processing requests for a more limited time period.

The Production Planner has to balance the various considerations when determining whether or not to create or update a PR.

Planning decisions are made on the basis of locally defined planning strategies for supporting the SCFs' data processing needs. The production planning tools are intended to be flexible enough in their design to support the particular planning and scheduling cycles of the operations organization at each DAAC.

Before planning production the Production Planner must coordinate with the Resource Planner to resolve all resource allocation issues. The Resource Planner notifies the Production Planner of the resources available for use in processing. Furthermore, the Production Planner may well have direct access to the Resource Plan.

The Production Planner prepares monthly and weekly production plans. In addition, the Production Planner develops a daily production schedule from the most current weekly plan. However, the first step in the planning process is creating production requests using the Production Request Editor.

26.12 PGE Registration and Test Data Preparation

The integration of science software with ECS requires that information about the Product Generation Executives (PGEs) be made known to the PDPS in its database. In addition, the PGEs themselves and the test files that they use (both input and output) need to be placed on the Data Server. These steps must be accomplished before the science software can be run and tested within the ECS.

The following procedures describe how to register a new PGE with ECS. This involves updating the PDPS database with information needed to plan, schedule, and run the PGE. The first step in the PGE registration process is to determine which ESDTs are needed for the PGE. You must Verify that an ESDT metadata ODL file exists for each ESDT or generate an ODL file. The next step in the process is to create a PGE metadata ODL file using the delivered PCF. Finally, additional operational information (resource requirements and runtime statistics) must be input into the PDPS database. This is the last step in the PGE registration process. The order in which these procedures are done is important and should be done as indicated. Please reference Appendix A for Examples of PGE and ESDT ODL Files for Each Instrument Team.

26.12.1 PGE ODL Preparation

This section describes how to prepare PGE ODL files. It is assumed that the SSIT Manager is running .

1 From the SSIT Manager, click on the **Tools** menu, then choose **PDPS Database** and then **PCF ODL Template**.

- An xterm with title “SSIT: Science Metadata ODL Template Creation” will be displayed.

2 At the program prompt **Configuration Filename (enter for default: ../../cfg/EcDpAtCreatODLTemplate.CFG)?**

- Press **Return** for the default configuration file

3 At the program prompt **ECS mode of operations?**, type ***mode***, press **Return,** or just press **Return** if the default shown is correct.

- The ***mode*** refers to the database used and will typically be **OPS** or **TS1**.

4 At the program prompt **PCF (Process Control file) to generate template from (including full path)?**, type ***PCFpathname*/*PCFfilename***, press **Return**.

- The ***PCFpathname*** is the full path name to the location of the PCF. If not specified, the directory from which the SSIT Manager was run will be assumed.
- The ***PCFfilename*** is the file name of the PCF.

5 At the program prompt **PGE name (max 10 characters)?**, type ***PGEname***, press **Return**.

- The ***PGEname*** is the name of the PGE that will be registered.

6 At the program prompt **PGE version (max 10 characters)?**, type ***PGEversion***, press **Return** or just press **Return** if the default shown is correct.

- The ***PGEversion*** is the version of the PGE that will be registered.

7 At the prompt **PGE Profile ID (0 for Null, max 999)?**, type 1 or any valid profile ID.

- After a brief time, the message “Successfully created ODL template file” should be displayed if the task was successful.
- The program will output a file with the filename **PGE_*PGEname#PGEversion#ProfileID*.tpl**.
- For example, if the PGE name was **PGE35**, and the version and profile ID were both **1** this output file will be named **PGE_PGE35#1#01.tpl**.

8 At the program prompt **Directory for output template file (including full path)?**, type **ODLtplPathname,** press **Return.**

- The **ODLtplPathname** is the full path to the directory for the output template file, e.g., **/usr/ecs/TS1/CUSTOM/data/DPS.**

9 At the program prompt **Hit return to run again, 'q <return>' to quit:**, press **Return** to repeat process with another **PCF** or type **q** and press **Return** to quit.

- The xterm will disappear.

10 At a UNIX prompt on an AIT Sun, type **cd *ODLtplPathname***, press **Return.**

- The ***SSITrunPathname*** is the full path to the directory from which the SSIT Manager was run, for example **/usr/ecs/TS1/CUSTOM/bin/DPS**. This will be the directory where the file **PGE_*PGEname#PGEversion#ProfileID*.tpl** will reside.

11 At a UNIX prompt on the AIT Sun, type **cp PGE_*PGEname#PGEversion#ProfileID*.tpl PGE_*PGEname#PGEversion#ProfileID*.odl**, press **Return.**

- The **PGE_*PGEname#PGEversion#ProfileID*.tpl** is the file name of the ODL template file created in step 8.
- The **PGE_*PGEname#PGEversion#ProfileID*.odl** is the file name of a copy that can be safely edited. This file name convention must be used.

12 At a UNIX prompt on the AIT Sun, type
mv PGE_*PGEname#PGEversion#ProfileID*.odl /usr/ecs/<mode>/CUSTOM/data/DPS/ODL

- This will place the ODL file in the directory from which the executable that populates the PDPS database will read. **PGE_*PGEname#PGEversion#ProfileID*.odl** is the file name of the copy created in step 11.

13 At a UNIX prompt on the AIT Sun, change the directory to the one in step above and type **vi PGE_*PGEname#PGEversion#ProfileID*.odl**, press **Return.**

- The **PGE_*PGEname#PGEversion#ProfileID*.odl** is the file name of the copy created in step 11.
- Any text editor may be used such as *emacs*. For example**, emacs PGE_PGE35#1#01.odl,** press **Return.**

14 In the file, add required metadata to the ODL template.

- For an explanation of what metadata is required, see file /usr/ecs/<mode>/CUSTOM/data/DPS/PGE_ODL.template.
- Note that the ShortNames typed into this file must each have a corresponding PDPS ESDT metadata ODL file.
- All objects corresponding to output ESDTs will automatically have the SCIENCE_GROUP and YIELD set during the generation of PGE ODL.
- All objects corresponding to output ESDTs will have an attribute "ASSOCIATED_MCF_ID". Place here the Logical Unit Number (LUN) listed in the PCF for the associated MCF listing.

- All objects corresponding to static input ESDTs must have the SCIENCE_GROUP set. Objects corresponding to *dynamic* input ESDTs should NOT have the SCIENCE_GROUP set.
- See Appendix E for an example of PCF and corresponding PGE ODL and ESDT ODL files.

15 Save the changes made to the ODL template file and exit the editor.

- The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, press **Return**.
- For other editors, refer to that editor's documentation

26.12.2 ESDT ODL Preparation

Assumption:

1. The PGE ODL file has been created and edited for the required PGE.

Follow the steps below to prepare ESDT ODL files for each ESDT required by the PGE.

1 Determine ShortName for required ESDTs corresponding to a Logical Unit Number (LUN) in the PGE ODL file.

2 At a UNIX prompt on an AIT Sun, type **ls /usr/ecs/<mode>/CUSTOM/data/DPS/ODL/ESDT_*ShortName#Version*.odl**, press **Return**.

- The **ESDT_*ShortName#Version*.odl** is the file name of the ESDT ODL file you are looking for where ***ShortName*** is the ESDT's ShortName and ***Version*** is the ESDT version. If a file for the desired ESDT is listed, then it has already been prepared and this procedure can be exited now.
- For example, if the desired ESDT has the ShortName MOD03 and version 001, type **ls /usr/ecs/TS1/S/CUSTOM/data/DPS/ODL/ESDT_MOD03#001.odl,** press **Return**.
- If the desired file is *not* listed, continue on to step 3.

3 At a UNIX prompt on the AIT Sun, type **cd *WorkingPathname***, press **Return.**

- The ***WorkingPathname*** is the full path name to a working directory for which the user has write permissions.
- For example, **cd /home/jdoe/working/**, press **Return**.

4 At a UNIX prompt on the AIT Sun, type **cp /usr/ecs/<mode>/CUSTOM/data/DPS/ESDT_ODL.template ESDT_*ShortName#Version*.odl**, press **Return**.

- For <mode> enter the mode you are working in, for example **OPS** or **TS1**.
- The **ESDT_*ShortName#Version*.odl** is the file name of the ESDT ODL file to be created.

- This command copies a template ESDT ODL file to the ESDT ODL file to be created. The template is well commented.
- For example, type **cp /usr/ecs/<mode>/CUSTOM/data/DPS/ESDT_ODL.template ESDT_MOD03#001.odl,** press **Return**.
- The **ESDT_*ShortName#Version*.odl** file naming convention *must* be observed.

5 At a UNIX prompt on the AIT Sun, type **vi ESDT_*ShortName#Version*.odl**, press **Return**.
- The **ESDT_*ShortName#Version*.odl** represents the file name of the ESDT ODL template file created in step 4.
- Any text editor may be used such as *emacs*. For example, **emacs ESDT_MOD03#001.odl**, press **Return**.

6 In the file, add required metadata to the ODL template.
- Use the internal documentation contained in the ODL file (from the original template) to aid in populating with metadata.
- Note that the ShortName specified within the file must match the ShortName of the file name itself.
- In addition, the ShortNames used in the PDPS PGE metadata ODL file must match the ShortNames in these files.

7 Save the changes made to the ESDT metadata ODL file and exit the editor.
- The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, press **Return**.
- For other editors, refer to that editor's documentation.

8 Next type **mv ESDT_*ShortName#Version.odl* /usr/ecs/<mode>/CUSTOM/data/DPS/ODL.**
- This will place the just created ESDT ODL file in the directory from which PDPS will read it.

9 Repeat steps 1 through 8 for each ESDT required by a particular PGE. When all ESDT metadata ODL files have been completed, continue on to next section.

26.12.3 Update the PDPS/SSIT Database with PGE Science Metadata

In order to update the PDPS Database with PGE metadata, the ESDT metadata ODL files must first be prepared for each ESDT required by the PGE. This section describes how to perform the next step, running the SSIT Science Update program.

Assumptions:

1. The SSIT Manager is running.

2. The directory used for containing the PDPS PGE metadata ODL files. Nominally, this is **/usr/ecs/<mode>/CUSTOM/data/DPS/ODL.**

Updating the PDPS Database with PGE Metadata

The directory used for containing the PDPS ESDT metadata ODL files can be accessed by the following commands:

1 telnet to (AITTL/DPS) **p0ais01** or a machine that matches the SSIT Manager host.

2 login: **ID**, password:

3 **setenv DISPLAY…….:0.0**

4 The directory used for containing the PDPS ESDT metadata ODL files. is **/usr/ecs/<mode>/CUSTOM/data/DPS/ODL**

5 From the SSIT Manager, click on the **Tools** menu, then choose **PDPS Database** and then **SSIT Science Metadata Update**.

- An xterm with title "SSIT: Science Metadata Database Update" will be displayed.

6 At the program prompt **Configuration Filename (enter for default: ../../cfg/EcDpAtRegisterPGE.CFG)?**

- Press **Return**.

7 At the program prompt **ECS mode of operation?**, type ***mode***, press **Return** or just press **Return** if the default shown is correct.

- The ***mode*** refers to the database used and will typically be **OPS** or **TS1.**

8 At the program prompt **PGE name (max 10 characters)?**, type ***PGEname***, press **Return**.

- The ***PGEname*** is the name of the PGE that will be registered. This name must match the PGE name specified.

9 At the program prompt **PGE version (max 10 characters1)?**, type ***PGEversion***, press **Return** or just press **Return** if the default shown is correct.

- The ***PGEversion*** is the version of the PGE that will be registered. This version must match the PGE version specified.

10 At the program prompt **PGE Profile ID (0-999, 0 means null)?** Type in a valid profile ID and press **Return,** or if already listed just press **Return.**

- The PDPS database will then be updated with the information contained in the file **PGE_*PGEname#PGEversion#ProfileID*.odl**

11 At the program prompt **Hit return to run again, q <return> to quit:**, press **Return** to update the PDPS database with another PGE ODL metadata file or type **q** and press **Return** to quit.

- If you make a mistake entering any values, press **Return** here; your previous entries are restored as defaults and you won't have to retype them.
- NOTE: If you make mistakes while editing the PGE and ESDT ODL files, you can run the ODL checker (Tools → PDPS Database → Check ODL) via the SSIT manager to locate any errors.

  ODL files must have been created to define the PGE to PDPS. Examples of the ODL files are under the data directory: PGE_ODL.template, ESDT_ODL.template, ORBIT_ODL.template, TILE_ODL.template and PATHMAP_ODL. A tool can be run to generate a template ODL file for the PGE from the SSIT Manager via Tools->PDPS Database->PCF Odl Template script. This then has to be populated with all information that can not be garnered from the PCF. The CheckOdl tool from the SSIT Manager via Tools->PDPS Database->Check ODL can be used to flag any errors in ODL before trying to put it in the database.

Sample of ESDT.odl files being established in ECS

```
home/emcleod/MODIS/STORE/PGE07
p0ais01{emcleod}10: ls
ESDT_MD10L2#001.odl     MOD_PR10            pge_cfg
ESDT_MD35L2#001.odl     MOD_PR10.mk         scf_cfg
ESDT_MOD02H#001.odl     PGE07.mk            script
ESDT_MOD03#001.odl      doc
p0ais01{emcleod}11: cp ESDT_MD10L2#001.odl ESDT_MD35L2#001.odl
ESDT_MOD02H#001.odl ESDT_MOD03#001.odl
/usr/ecs/OPS/CUSTOM/data/DPS/ODL/
```

26.12.3.1 Alternative Tool for SSIT Metadata Update

1 Source the buildrc file for the mode in which you are working (*source .buildrc)*. **/usr/ecs/<MODE>/CUSTOM/utilities,** Note that this only has to be done once per login. Then **(*cd* /usr/ecs/<MODE>/CUSTOM/bin/DPS)**

- The tool can also be executed by being in the **/usr/ecs/<MODE>/CUSTOM/bin/DPS** and executing **EcDpAtDefinePGE..**
- Shell script prompts user for information.

2 Enter in the location of the configuration file **(../../cfg/EcDpAtRegisterPGE.CFG).**

3 Enter the MODE of operation (<MODE>).

4 Enter name of PGE (it must match what is in the PGE ODL file).

5 Enter the version of the PGE (it must match what is in the PGE ODL file).

6 Enter the Profile ID (it must match what is in the PGE ODL file). Note that the ODL file for the PGE must have the of: PGE_<PGE NAME>#<PGE VERSION>#<PROFILE ID>.

- Each ODL file is displayed as it is processed. A good status message should be displayed as a result. Information about the PGE (inputs and outputs, Production Rules, etc) should be entered in the Database.

26.12.3.2 Examples of PGE and ESDT ODL Files for Each Instrument Team

This section is taken from the **Green Book, 162-TD-001** and materials included as Appendix A to this document. Depicted are examples of ODL files in SSI&T activities. Then, examples of specific ODL files are listed by instrument (ASTER, MISR or MODIS).

Template ODL Files

There are five Template ODL files listed therein. The specific or tailored ODL files listed were derived from these templates by appropriate editing and filling-in of values. The three ODL Template files listed reside, on the AIT Sun host, at **/usr/ecs/<mode>/CUSTOM/data/DPS** . They are

- PGE_ODL.template
- ESDT_ODL.template
- ORBIT_ODL.template
- PATHMAP_ODL.template
- TILE_ODL.template

Example of a successful PDPS Science Metadata Update:

PDPS/SSIT SCIENCE Metadata Database Update **
Configuration filename? (enter for default: ../../cfg/EcDpAtRegisterPGE.CFG)
ECS Mode of operations? (enter for default: OPS)
OPS
PGE name (max 10 characters)?
PGE07
PGE version (max 10 characters)?
001
PGE Profile ID (0-999, 0 means null)? (enter for default: 1)
1
Warning: Could not open message catalog "oodce.cat"

```
EcDpAtRegisterPGE: Process Framework: ConfigFile
../../cfg/EcDpAtRegisterPGE.CFG  ecs_mode OPS
' PGE profile id = '1' ...
Do you wish to overwrite the previous PGE PGE07( (y)es or (n)o):
y
FILES PROCESSED
:  PGE SCIENCE ODL file =/usr/ecs//OPS/CUSTOM/data/DPS/ODL/PGE_PGE07#0#001.odl
   ESDT SCIENCE ODL file = /usr/ecs//OPS/CUSTOM/data/DPS/ODL/ESDT_MOD02H#001.odl
   ESDT SCIENCE ODL file = /usr/ecs//OPS/CUSTOM/data/DPS/ODL/ESDT_MD35L2#001.odl
   ESDT SCIENCE ODL file = /usr/ecs//OPS/CUSTOM/data/DPS/ODL/ESDT_MOD03#001.odl
   ESDT SCIENCE ODL file = /usr/ecs//OPS/CUSTOM/data/DPS/ODL/ESDT_MD10L2#001.odl
********** Update of PDPS/SSIT database with  PDPS SCIENCE metadata SUCCESSFUL **********
Hit return to run again, 'q <return>' to quit:
```

26.12.4 Operational Metadata

The SSIT version of the PDPS database is initialized and updated with SSIT Operational Metadata so that the Planning and Processing Subsystem can schedule and run PGEs. Here, PDPS Operational Metadata refers to PGE information that is supplied to the DAAC/SSIT Operator and may change frequently.

The operator enters this data directly into the SSIT Operational Metadata Update GUI. The program then writes the data directly to the SSIT version of the PDPS database.

Before running the SSIT Operational Metadata Update from the SSIT Manager, you must first update the PDPS with SSIT Science Metadata. In addition, to get initial PGE Performance data that will be entered into the GUI, you need to run the profiling utility, EcDpPrRusage on the PGE or have the information on profiling provided. See section 26.12.2.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The required UNIX environment variables have been set.
2. The Science metadata has been updated to the PDPS database for this PGE.

To update the SSIT version of the PDPS database with operational metadata, execute the steps that follow:

1 From the SSIT Manager, click on the **Tools** menu, then choose **PDPS Database** and then **SSIT Opnl Metadata Update**.
- The PDPS/SSIT Database Update GUI, **"dpAtOpDbGui"** will be displayed.

2 Click on the radio button labeled **NEW PGE** in the lower left quadrant.
- The PGE that you are working on should appear in the subwindow labeled **PGE Names** along with its version number in the subwindow labeled **PGE Versions.**

3 In the subwindow labeled **PGE Names**, click on a PGE name. Then in the subwindow labeled **PGE Versions**, click on the PGE version for that PGE. Then click on the button labeled **EDIT**.

- The PGE name and version will be highlighted when you click on them.
- The page tabs **PROFILE**, **RUNTIME**, and **ESDT** will change from gray (indicating disabled) to black (indicating enabled).
- To see the contents of PGE Metadata, click on the button labeled **DISPLAY** and then click on the button labeled **DONE.**
- If the PGE name and/or version does (do) not appear in the lists, it means that updating of PDPS database with PGE metadata was not successful.

4 Click on the **PROFILE** page tab.

- The Profile page will be displayed.

5 In the fields under the label **Performance Statistics**, enter the information specified.

- In the field labeled **Wall clock time**, enter the amount of wall clock time it takes for one execution of the PGE, in seconds. The tab **PROFILE** will change from black (indicating enabled) to red (indicating database needs to be updated by APPLY button).
- In the field labeled **CPU time (user)**, enter the so-called *user* time of the PGE, in seconds. This value should come from profiling the PGE .
- In the field labeled **Max memory used**, enter the maximum amount of memory used by the PGE, in megabytes (MB). This value should come from profiling the PGE .
- In the fields labeled **Block input ops** and **Block output ops**, enter the integer number of block inputs and block outputs, respectively. These values should come from profiling the PGE .
- In the field labeled **Swaps**, enter the integer number of page swaps from the PGE. This value should come from profiling the PGE .
- In the field labeled **Page faults**, enter the integer number of page faults from the PGE. This value should come from profiling the PGE .

6 In the fields under the label **Resource Requirements**, enter the information specified.

- In the field labeled **Runtime Directory Disk Space**, enter the maximum amount of disk used by the PGE during execution, in megabytes (MB), for staging MCF files, system PCF file, Profile file and generating log files. Typically this is 20 MB.
- Click on one of the two radio buttons labeled **Proc. String** and **Computer Name** (if not already clicked on).
- A list of processing strings should appear in the scrollable window to the left of the two radio buttons **Proc. String** and **Computer Name**. Nominally, only one item should be listed and should be highlighted.
- In the field labeled **Number of CPUs**, the number 1 should be typed.

7 In the field labeled **Local filename of top level shell**, type in the appropriate top level executable file name within a science software executable package.

8 In the field labeled **SGI Application Binary Interface (ABI)**, click on the selection appropriate for your PGE.

9 Once the fields on the **PROFILE** page have been completed, click on the **APPLY** button.

- This will update the PDPS database with the information just entered. The tab **PROFILE** will change from red (indicating database needs to be updated) to black (indicating enabled).
- An information box will be displayed, click on **Ok**.
- To start over, click on the **RESET** button. This will clear all fields.

26.12.5 SSIT Operational Metadata Update GUI

The SSIT version of the PDPS database is initialized and updated with SSIT Operational Metadata so that the Planning and Processing Subsystem can schedule and run PGEs. Here, PDPS Operational Metadata refers to PGE information that is supplied to the DAAC/SSIT Operator and may change frequently.

The operator enters this data directly into the SSIT Operational Metadata Update GUI. The program then writes the data directly to the SSIT version of the PDPS database. The SSIT Operational Metadata Update GUI is used to view or update the following operational parameters for a particular PGE:

- Performance parameters for the PGEs.
- Resource parameters for the PGEs.
- PGE user-defined static parameter.
- View the PGE science metadata file.

26.12.6 Test Data Preparation and Insertion of Data Granules

This section describes how to prepare test data for use by registered PGEs. When PGEs are first delivered to the DAAC and registered within the PDPS, they will typically be run in isolation. That is, they will be run without any PGE dependencies. For this testing to be possible, test input data granules required by the PGE need to be pre-Inserted to the Data Server.

Data granules can be *dynamic* or *static*. Dynamic data granules are those whose temporal locality differs for each instance of the granule. Examples of dynamic granules are Level 0, Level 1, and Level 2 data sets. Static data granules are those whose temporal locality is static over long periods of time. Examples of static granules are calibration files that may only change with a new version of a PGE. For any granule to be Inserted to the Data Server, a Target MCF is needed (also known as an ASCII metadata ODL file or a **.**met file).

In the actual production environment, a Target MCF is produced by the PGE during execution. Thus, the data granule can be inserted. In isolation testing of a PGE, however, the inputs needed by it, will not have been inserted by a previous PGE in the chain. This Insertion must be done manually. The next two sections describe how to use the Source MCF for a dynamic data granule to create a Target MCF. and then describes how to do the Insert. In this way, a dynamic data granule can be inserted to the Data Server as if a PGE had produced it.

26.12.6.1 Generating a Metadata Configuration File (Source MCF)

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The SSIT Manager is running.
2. ESDTs are installed onto the **Science Data Server.**

To Generate the Metadata Configuration File (Source MCF) for the input and output ESDTs, execute the steps that follow.

1 From the SSIT Manager, click on the **Tools** menu, then choose **Data Server** and then **Get MCF.**

- An xterm in which EcDpAtGetMCF is running will be displayed as SSIT: Acquire MCF..
- Alternatively, the same tool can be invoked by typing at a UNIX prompt on an AIT Sun **EcDpAtGetMCF.sh**, press **Return**.

2 At the program prompt **Configuration Filename (default *defaultConfigFile*)?**

- Type in **../../cfg/ *defaultConfigFile*** and press **Return**.
- The ***defaultConfigFile*** will be replaced by the full path name and file name of the default configuration file. The file name will be **EcDpAtGetMCF.CFG** where *daac* will be replaced by one of {GSFC, EDC, LARC, NSIDC}.

3 At the program prompt **ECS mode of operation (enter for default: def*aultMode*)?**, type ***mode***, press **Return** or just press **Return** if the default shown is correct.

- The ***mode*** refers to the database used and will typically be **TS1**.

4 At the program prompt **ESDT Short Name?**, type ***ESDT ShortName***, press **Return**.

- The ***ESDTShortName*** is the name of the ESDT that the EcDpAtGetMCF tool will use to generate the MCF.

5 At the program prompt **ESDT Version?**, type ***ESDTversion***, press **Return** or just press **Return** if the default shown is correct.

- The ***ESDTversion*** is the version of the ESDT.

6 At the program prompt **Directory to receive MCF (must be full path)?**, type ***MCFpathname***, press **Return**.

- The ***MCFpathname*** is the full path name to the location where the source MCF will be placed. For example, /home/jdoe/ssit.

7 To the final prompt **Hit return to run again, 'q <return> to quit:**, press **Return** to generate another Source MCF or type **q** and press **Return** to quit.

- If you make a mistake entering any values, press **Return** here; your previous entries are restored as defaults and you won't have to retype them.

Example of a successful installation of a Source MCF:

Configuration filename? (enter for default: ../../cfg/EcDpAtGetMCF.CFG)
ECS Mode of operations?
OPS
ESDT Short Name?
MOD03EM
ESDT Version?
0
Directory to receive MCF? (must be full path)
/home/emcleod/MCF/
Warning: Could not open message catalog "oodce.cat"
EcDpAtGetMCF: Process Framework: ConfigFile ../../cfg/EcDpAtGetMCF.CFG
 ecs_mode
 OPS
incomplete group entries in the configfile, using default G1
Request for MCF successful for:
  ESDT name = 'MOD03EM'
  ESDT version = '0'
  directory = '/home/emcleod/MCF/'
Hit return to run again, 'q <return>' to quit:

26.12.7 Creating a Target MCF (.met) for a Dynamic/Static Granule

A Target MCF file for a corresponding data granule can be created based on the information provided in the Source MCF file and the involved science software package (PGE).

In standalone or isolation testing of a PGE, the inputs it needs will not have been Inserted by a previous PGE in the chain. This Insertion must be done manually. A Target MCF file for a corresponding data granule is required to run a standalone PGE. This way a dynamic data granule can be Inserted to the Science Data Server as if a PGE had produced it.

The following steps can be used to obtain .met files for standalone PGE runs.

1 For all dynamic data granules, try to locate a .met file from an output of a previous PGE. Usually the input to a PGE is the output of some previous PGE. Hence, one can use the relevant documents from the instrument team to obtain such information on the required

PGE. Once a .met file is available, all you need to do is edit the timestamp for your run of the PGE.

2 Use HDF_EOS-view from the SSIT manager to take a look at the header of the HDF data granules. The header contains information on the .met file.

26.12.7.1 Creating a Metadata File for a Static Granule

1 At the UNIX prompt on the AIT Sun, type cd WorkingPathname, then press the **Enter** key.

- Example: **cd /usr/ecs/{MODE}/CUSTOM/data/DPS/ODL/**
- The ***WorkingPathname*** *is* the full path name of the working directory containing the template metadata ODL file.

2 At the UNIX prompt on the AIT Sun, type **cp *StaticODLmet.tpl filename*.met**, then press the **Enter** key.

- The ***StaticODLmet.tpl*** is the file name of the template Target MCF.
- The ***filename*.met** is the file name of the Target MCF for this static file. The file name extension must be **.met**.
- This command will copy the template Target MCF to ***filename*.met**. For example, type **cp StaticODLmet.tpl CER11T.mcf.met**, then press the **Enter** key.

3 At a UNIX prompt on the AIT Sun, type **vi *filename*.met**, then press the **Enter** key.

- This command invokes the *vi* editor and reads in the Target MCF created above.

4 Edit the Target MCF with the specific information for the static data granule to be Inserted. The following guidelines should be followed when editing on the template MCF:

- The value for the ShortName object should be filled out with proper instrument name.
- The value for the Version ID object should be filled out with the proper version number.
- In the **INFORMATIONCONTENTCONTAINER** object enter the following:
  - The value for the **PARAMETERNAME** object of the class **"1"** should be filled out with the name of static data file.
  - The value for the **PARAMETERVALUE** object of the class **"2"** should be filled out based on the following guideline:

    If the data granule is a coefficient file, a **"C"** followed by a numerical number **n (n=1,2,….)** will be used. Here n stands for the number of the coefficient file.

If the data granule is an MCF file, an “M” followed by a numerical number **n (n=1,2,….)** will be used. Here n stands for the number of the MCF file.

5 Save the changes made to the Target MCF (***filename*.met**) and exit the editor.

- The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, then press the **Enter** key.

26.12.8 Inserting Static Data Granules into the Data Server

Inserting a Static Data File:

The following Servers/Services must be up and operational:

- **Science Data Server, Storage Management.**

The following must have occurred between those Servers/Services:

- The ESDT of the static file must have been installed at the Data Server.

What the user must do before trying SSIT functionality:

- Create a metadata file for the static file to insert. To do this, an MCF (See “Getting an MCF in this section”) must be gotten from the Data Server for the ESDT of the file to insert. Mandatory fields are filled into the MCF, creating a metadata file.
- If the tool is NOT run from the SSIT Manager then go to the executables directory **(cd /usr/ecs/<MODE>/CUSTOM/utilities)**
- Source the buildrc file for the mode in which you are working **(source .buildrc).** Note that this only has to be done once per login.
- If the tool is NOT run from the SSIT Manager then go to the executables directory (**cd /usr/ecs/<MODE>/CUSTOM/bin/DPS)**

1 From the **SSIT Manager** choose **Tools** menu and then **Data Server** submenu. Choose **Insert Static File.**

- The tool can also be executed by being in the /usr/ecs/<MODE>/CUSTOM/bin/DPS and executing EcDpAtInsertStatic
- Shell script prompts user for information.

2 Enter in the location of the DpAtInsertStaticFile configuration file **(../../cfg/EcDpAtInsertStaticFile.CFG).**

3 Enter the MODE of operation **(<*MODE*>).** At the program prompt **mode (default ops)?**, or press **Enter** to take default.

4 Enter the short name of the ESDT (for the static file). This value is in the PDPS database under the PlDataTypeMaster table and must be in the PGE ODL file.

- At the program prompt **ESDT name?** type ***ESDTShortName***, then press the **Enter** key. For example type: **MOD02LUT.**

5 Enter the version of the ESDT for the static file. This value is also in the PDPS database under the PlDataTypeMaster table and must be in the PGE ODL file.

- At the program prompt **PGE version (default 1)?**, type ***PGEVersion***, then press the **Enter** key.
- The ***PGEVersion*** must match exactly the PGE version entered into the PDPS for this PGE.

6 Enter the science group for this static (this will be from the ODL created during Populating the PGE information in the Database).

- At the program prompt **Science group for Static file(one of{C,L,D,O} followed by a 4 digit number)?**, type ***ScienceGroupID***, then press the **Enter** key.
- The ***ScienceGroupID*** is an identifier used to define the file type as a coefficient file, a lookup table file, or a MCF. It distinguishes static granules of different types that share the same ESDT. For instance, for a coefficient file, use **C*n***, where number ***n*** could be **0, 1, 2…;** this number *n* needs to be matched with the number *n* in the PGE_PGEName#Version.odl file. For an MCF. For example, type **C001**, press **Return.**
- The Science Group ID must match what was edited into the PGE metadata ODL file for that PCF entry.

7 At the program prompt **Is there more than one data file for this Static (Y = Yes, N = No)? (enter for default: N).** If there is only one data file, press **Return** and go to next step. If there are more than one data files, type **Y,** press **Return** and go to step 10.

8 At the program prompt **Single Static Filename to Insert (including FULL path)?**, type ***pathname*/*GranuleFileName***, press **Return**

- The ***pathname*/*GranuleFileName*** is the full path name and file name of the static data granule to be Inserted. For example, type **/home/MODIS/PGE10/MOD_PR28/coeff/emissivity.dat**, press **Return**.

9 At the program prompt **Associated ASCII Metadata Filename to Insert (including FULL path).** Type ***pathname/GranuleFileName.met*)?**, press **Return**.

- The ***pathname*/*GranuleFileName.met*** is the full path name and file name of the .met file for the associated static data granule to be Inserted. For example, type **/home/MODIS/PGE10/MOD_PR28/MOD28LUT.met** press **Return**.

10 At the program prompt **Directory where all data files and .met file exist (FULL path)?** Type ***pathname*** press **Return**.

- where ***pathname*** is the full path of the directory where all data files and .met file exist.
- Note for a multi-file granule, the data files and .met file should be placed in the same working directory.

11 At the program prompt **Name of MFG file (enter to end list)?** Type in the ***GranuleFileName,*** one at a time and press **Return.** To end the list press **Return.**

- Where ***GranuleFileName*** is the names of the multi-file granules.

12 At the program prompt **Associated ASCII Metadata Filename to Insert?** Type ***GranuleFileName.met***, press **Return.**

- Where ***GranuleFileName.met*** is the name of one **.met file** that is used with all data granules in the even of a multi-file granule.
- The dynamic data granule will be inserted to the Data Server. For reference, the Data Server Universal Reference (UR) will be printed on the screen.

13 At the program prompt **Hit return to run again, 'q <return>' to quit:** type **q** and press **Return** to quit or just press **Return** to insert additional dynamic granules.

- If continuing, repeat steps 2 through 9.

26.12.9 Inserting Dynamic Data Granules into the Science Data Server

In order for dynamic data files to be used both during the SSI&T and in production, this file must exist in the Data Server and be accessible by the local machine. A program called the Insert Test Dynamic File can be used for inserting a dynamic data granule into the Data Server.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The ESDTs have been installed on the Data Server.
2. The Target MCF for this data granule has been created for the Insert.

To Insert a dynamic granule to the Data Server, execute the following steps:

1 From the SSIT Manager, click on the **Tools** menu, then choose **Data Server** and then **Insert Test Dynamic**.

- An xterm with title "SSIT: PGE Test Dynamic Input File Insertion" will be displayed.

2 At the program prompt **Configuration filename? (enter for default: ../../cfg/EcDpAtInsertTestFile.CFG)**, press **Return**.

3 At the program prompt **ECS Mode of operations?**

- Type in the **<mode>** you are working in. For example, **TS1** or **OPS.** Press **Return.**

4 At the program prompt **ESDT short name for the file(s) to insert?** type ***ESDTShortName***, press **Return**

- The ***ESDTShortName*** is the ShortName of the ESDT descriptor file corresponding to this granule to be inserted. For example, type **MOD021KM** press **Return**.

5 At the program prompt **ESDT Version for the file(s) to insert?** Type in the ESDT version and press **Return.**

6 At the program prompt **Is there more than one data file to this Dynamic Granule (Y = Yes, N = No)? (enter for default: N)?** If there are no multi-files for this ESDT, press **Return** and go to step 7. If there are more than one file for this granule go to step 9.

7 At the program prompt **Single Filename to Insert? (including FULL path)** type ***pathname*/*GranuleFilename***, press **Return.**

- The ***pathname*/*GranuleFileName*** is the full path name and file name of the data granule to be inserted. For example, type **/home/MODIS/PGE10/MOD021KM.A1996217.0014.002.hdf**, press **Return**.

8 At the program prompt **Associated ASCII Metadata Filename to Insert (including FULL path)** , Type ***pathname*/*GranuleFileName.met*** and press **Return**.

- ***pathname*** is full name of the path and ***GranuleFileName.met*** is the name of the associated .met file. For example, **/home/MODIS/PGE10/MOD021KM.met**
- The dynamic data granule will be inserted to the Data Server. For reference, the Data Server Universal Reference (UR) will be printed on the screen.

9 At the program prompt **Directory where all data files and .met file exist (FULL path)?** Type ***pathname*** press **Return**.

- where ***pathname*** is the full path of the directory where all data files and .met file exist.
- Note for a multi-file granule, the data files and .met file should be placed in the same working directory.

10 At the program prompt **Name of MFG file (enter to end list)?** Type in the ***GranuleFileName,*** one at a time and press **Return.** To end the list press **Return.**

- where ***GranuleFileName*** is the names of the multi-file granules.

11 At the program prompt **Associated ASCII Metadata Filename to Insert?** Type ***GranuleFileName.met***, press **Return.**

- where ***GranuleFileName.met*** is the name of one .met file that is used with all data granules in the even of a multi-file granule.
- The dynamic data granule will be inserted to the Data Server. For reference, the Data Server Universal Reference (UR) will be printed on the screen.

12 At the program prompt **Hit return to run again, ‘q <return>’ to quit:** type **q** and press **Return** to quit or just press **Return** to insert additional dynamic granules.

- If continuing, repeat steps 2 through 8.

Example of a successful insertion of a Dynamic Input Data Granule into the Data Servers:

PGE Test Dynamic Input File Insertion **
Configuration filename? (enter for default:
../../cfg/EcDpAtInsertTestFile.CFG)
ECS Mode of operations? (enter for default: OPS)
OPS
ESDT name
MOD02H
ESDT Version (enter for default: 1)
0
Staged Filename to Insert? (including FULL path)
/home/emcleod/MCF/MOD02HKM.A1997217.1730.002.hdf
Associated ASCII Metadata Filename to Insert? (including FULL path)
/home/emcleod/MCF/MOD02H.met

Warning: Could not open message catalog "oodce.cat"
EcDpAtInsertTestFile: Process Framework: ConfigFile
../../cfg/EcDpAtInsertTestFile.CFG ecs_mode OPS
incomplete group entries in the configfile, using default G1
Trying to make a request to [MDC:DSSDSRV]
incomplete group entries in the configfile, using default

Trying to make a request to [MDC:DSSDSRV]
incomplete group entries in the configfile, using default

Insert to Data Server successful:
ESDT Version = ’0’

staged file = ’/home/emcleod/MCF/MOD02HKM.A1997217.1730.002.hdf’
metadata file = ’/home/emcleod/MCF/MOD02H.met’
Inserted at UR:
’UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[MDC:DSSDSRV]:16:SC:M
OD02H:1757’
Hit return to run again, ’q <return>’ to quit:

26.12.10 Science Server Archive Package (SSAP)

The SSAP is used to provide a record of the science software, documentation, and other related files stored at the DAAC. The SSIT SSAP GUI provides a method for grouping required data about a PGE.

The SSAP is not to be confused with the Delivered Algorithm Package (DAP) received from the SCF. Much of what is in the DAP will make it into the SSAP. The key difference is that SSAP data is prepared after initial testing of the science software and will include data that reflects site integration as well as fixes required for performance at the DAAC.

The SSAP is made up of 2 different data types. The first data type is the Algorithm Package, which contains metadata (name of the PGE, name of the instrument, date accepted, etc.) about the SSAP. The second data type is the source code, documentation, and test data that will be stored as a SSAP, with its own metadata in addition to the files. SSAP components such a source code will be tarred to retain the directory structure.

The executables and static files are stored separately from the SSAP and will have their own data types (ESDTs).

Before continuing on we recommend that a review be made of the latest SSAP documentation contained in the Internal Interface Control Document for the EMD Project, 313-CD-001. Here you will find step by step procedures that cover the various Thread Components you may encounter.

What follows is a list of items in the SSAP. See also the Core Metadata model under DAP for a graphical representation of the SSAP.

The following make up an SSAP:

- Documentation
  - Delivery Memo
  - Summary Information for each PGE.
  - System Description Document (SDD).
  - Operations Manual
  - Processing Files Description Document
  - Test Plans (these include the test cases)
  - Scientific documents
  - Interface Definition Document
  - Detailed Performance Testing Results
  - Detailed design/implementation documents
  - COTS User or Programmer Guides
- Software & Control files:
  - Science software source code (including make files & scripts)
  - Testing software source code (including make files & scripts)
  - Test Data Input (this may only be the UR for this)
  - Expected Test Output

- Coefficient Files
- Process Control File
- Metadata Configuration File
- ODL files. These define the PGE and its related Data Types to the PDPS database. They don't currently have official names.

- Other files:
  - A change log created by the SSAP GUI to track changes to the SSAP.

The following is a list of tools and/or assumptions:

1. The SSIT Manager is running.
2. The PGE has been successfully built with the SCF and DAAC version of the Toolkit.

26.12.10.1 Creating an SSAP

The following Servers/Services must be up and operational:

- **Science Data Server, Storage Management.**

What the user must do before trying SSIT functionality:

1 From the SSIT Manager choose **Tools** menu and then **Data Server** submenu. Choose **SSAP Editor.**

- The GUI starts. Note that it will first query Data Server for a list of SSAPs that have previously been created. This list will appear in the window at the center (if any SSAPs already exist).   Current SSAP field will be blank, and only Refresh and Create buttons will be active. All three tabs (Main, Files, and Metadata) will be active.
- The SSAP GUI will be displayed.

2 Click on the **Create** to create a new SSAP.

- The **Create SSAP** window appears. If no OK button is visible, resize the window such that the OK button is visible.

3 Enter the name of the SSAP in the first field . Enter SSAP version in the second field. Note that version has a limit of 20 characters.

4 Click **OK** and the window disappears

- On the main GUI, the SSAP created (what was entered in the step above) will appear. Current SSAP is now set to that value. All buttons are now active.

5 To set up the SSAP components, click on the **File List** tab.

- The File List Tab displays files in the local directory to the left and files in the selected SSAP component to the right. On the bottom left is a directory listing and a method to move through the directory tree on the local machine. Delete and Reset buttons—both active – are to the right.

6 To select a file in the left column, click on the **File Type** button, highlight a file (or files) and click on the **Add** arrow button to add the files..

- The files to be added will be displayed in the right column.
- To change directories (and thus add files from other directories to the SSAP component), click on the listing in the window on the bottom left of the GUI. The ".." is to go up one directory level. A single click with move to the directory chosen and change the display to show the directories under the new current directory. Note that the list of files in the upper left window changes to show the files within the current directory.

7 To add metadata for the new SSAP, select the **Metadata** tab.

- The Metadata window will be displayed.
- The Metadata Tab displays the metadata for the new SSAP. Only the Name and Version will be filled in automatically. The rest of the fields will have default information. While the SSAP can be submitted with the default information, it is wise to fill in valid values. To change a value:

8 To change the default information, click on the **Edit Assoc Collections** button.

- The **Edit Associated Collections** window displays a list of associated collections and fields for the entry of new ShortNames and Versions.

9 Enter a ShortName, and version (of the ESDT that has been installed in the Data Server) - must be eight or fewer characters. Note that the Data Server will verify if the Shortname exists.

10 Enter the version (of the installed ESDT). Then select the **OK** button. Select **Done** to close the window.

11 To save the updated metadata, click **Save** on the **Metadata** tab**.**

12 To get back to the **Main** tab**,** select the **Main** tab button.

13 To submit the new SSAP to the Data Server, select the **Submit** button.

- When the SSAP has been submitted, the **SSAP Successfully inserted to the Data Server** prompt will appear.

The SSAP Editor is covered in more detail in the 313-EMD-001, Internal Interface Control Document for the EMD Project, in the section where Insert Thread Components are covered. Information on the SSAP Editor is also available in 609-EMD-001 Operations Tools Manual for the EMD Project.

26.12.11 Inserting an SSAP into PDPS

This procedure describes how to insert an SSAP into PDPS.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. All required Servers are up running.
2. SSI&T is up and running. (See section 6 how to bring up SSIT Manager)
3. The C shell (or a derivative) is the current command shell.

FORCHECK is available only on the AIT Suns.

To create the SSAP, execute the procedure steps that follow:

1 If not already on an AIT Sun, log onto one from your machine.

2 Launch the SSIT Manager

3 From the SSIT Manager choose *Tools* menu and then *Data Server* submenu. Choose *SSAP Editor*.
- The GUI starts. Note that it will first query Data Server for a list of SSAPs that have previously been created. This list will appear in the window at the center (if any SSAPs already exist). Current SSAP field will be blank, and only Refresh and Create buttons will be active. All three tabs (Main, Files, and Metadata) will be active.

4 Click on *Create* to create a new SSAP.
- The Create SSAP window appears. If no OK button is visible, resize the window so that the OK button is visible.

5 Enter the name of the SSAP in the first field.

6 Enter the SSAP version in the second field. Note that version has a limit of 20 characters.

7 Click *OK* and the window disappears.
- On the main GUI, the SSAP created (what was entered in the step above) will appear. Current SSAP is now set to that value. All buttons are now active.

8 Click on the *File List tab* to set up SSAP components.

- The File List Tab displays files in the local directory to the left and files in the selected SSAP component to the right. On the bottom left is a directory listing and a method to move through the directory tree on the local machine. Delete and Reset buttons -- both active – are to the right.

9 Click on the *File Type* button to select the SSAP component to manipulate.

10 Choose one of the menu items.

11 Select a file (or files) from the left window to add to the component.

12 Click the *Add Arrow* button to add the files. They will appear in the right window because they are now part of that SSAP Component.

13 Now select the *Metadata* tab to set the metadata for the new SSAP.

- The Metadata Tab displays the metadata for the new SSAP. Only the Name and Version will be filled in automatically. The rest of the fields will have default information. While the SSAP can be submitted with the default information, it is wise to fill in valid values. To change a value, Click the mouse in the field you wish to change and type in a new value. For dates click in the first box or use the up/down arrows to move the date up or down. When finished entering a date, click the *OK* button. For text fields just hit the *Enter* key. The button marked "Edit Assoc. Collections" on the bottom of the window must be hit and an Associated Collection entered for the SSAP.

14 Click the *Edit Assoc. Collections* button.

- The Edit Associated Collections window displays a list of associated collections and fields for the entry of new ShortNames and Versions (which make up an Associated Collection).

15 Enter a shortname (of an ESDT that has been installed in the Data Server) — must be eight or fewer characters. Note that the Data Server will verify if the Shortname exists.

16 Enter the version (of the installed ESDT).

17 Click *OK* and the new entry to the collection should appear in the window.

18 Click *Done* to close the window.

19 Click on the Metadata tab.

20 Click *Save* to save the updated metadata.

21 Click *Main tab* to get back to the Main tab.

22 Click *Submit* to send the new SSAP to Data Server. When finished, a message should pop up that says “SSAP Successfully inserted to the Data Server”.

26.12.12 Updating a Science Software Archive Package (SSAP)

The following Servers/Services must be up and operational:

- **Data Server, Storage Management.**

The following must have occurred between those Servers/Services:

- An SSAP must have already been inserted to the Data Server.
- What the user must do before trying SSIT functionality:
- The SSAP Editor has been used to insert an SSAP to the Data Server.

What must be done via SSIT tools:

- If SSAP Editor is not running, use the directions from the first 2 paragraphs of (Creating an SSAP) to bring up the SSAP GUI. Note that the added SSAP should appear in the window of the Main tab.

If the SSAP Editor is already running, the added SSAP should appear in the window of the Main tab.

1 Click on **added SSAP** in the main display.

2 Click on the **Metadata tab** to update the SSAP.

- The Metadata Tab displays the metadata for the SSAP. All fields will be set to the values entered when the SSAP was created, and the Algorithm Name field will be grayed out (because it may not be updated). If you want to create a new SSAP from an existing one, go back to the Main tab and hit the Create With button.

3 Click on the Algorithm Version field (currently called Algorithm Description) and enter a new version (different from what is in the field when the tab is clicked).

4 Update any other fields that you wish to change. You can even add a new Associated Collection by clicking on the Assoc Collection button and following the steps described in Creating an SSAP.

5 Before you leave the Metadata tab, click Save to save the updated metadata.

6 Click on the File List tab to set up new SSAP components.

- The File List Tab displays files in the local directory to the left and files in the selected SSAP component to the right. On the bottom left is a directory listing and a method to move through the directory tree on the local machine. Delete and Reset buttons—both active –are to the right.

7 Click on the File Type button to select the additional SSAP component to manipulate.

- Choose one of the menu items.
- Select a file (or files) from the left window to add to the component.
- Click the Add Arrow button to add the files. They will appear in the right window because they are now part of that SSAP Component.
- Click Main to get back to the Main tab.
- On the Main tab:
- Click Submit to send the new SSAP to Data Server. When finished, a message should pop up that says "SSAP Successfully inserted to the Data Server".

The Science Software Archive Package (SSAP) is a grouping of science software, documentation, and other related files that is stored at the DAAC. For a discussion of the SSAP and its contents, see Section 26.12.11, Inserting an SSAP into PDPS.

26.12.12.1 Updating an SSAP

This procedure describes how to update an existing SSAP in PDPS.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. All required Servers are up and running.
2. An SSAP must already have been inserted into the Data Server.
3. The C shell (or a derivative) is the current command shell.
4. FORCHECK is available only on the AIT Suns.

To update the SSAP, execute the procedure steps that follow:

1 If not already on an AIT Sun, log into one from your machine.

2 Launch the SSIT Manager. (See section 6).

3 From the SSIT Manager choose Tools menu and then Data Server submenu. Choose SSAP Editor.

- The GUI starts. Note that it will first query Data Server for a list of SSAPs that have previously been created. This list will appear in the window at the center (if any SSAPs already exist--if not, one will, of course, have to be created before the remainder of this procedure can be performed). Current SSAP field will be blank, and only Refresh and Create buttons will be active. All three tabs (Main, Files, and Metadata) will be active.
- There is currently missing functionality in the SSAP Editor, so before updating the new SSAP you must hit the Refresh button to refresh the data about the new SSAP.

4 Click on the Metadata tab to update the SSAP.

- The Metadata Tab displays the metadata for the SSAP. All fields will be set to the values entered when the SSAP was created, and the Algorithm Name field will be grayed out (because it may not be updated). If you want to create a new SSAP from an existing one, go back to the Main tab and hit the Create With button.

5 Click on the Algorithm Version field (currently called Algorithm Description) and enter a new version (different from what is in the field when the tab is clicked).

6 Update any other fields that you wish to change. You can even add a new Associated Collection by clicking on the Assoc. Collection button and following the steps described in creating an SSAP.

7 Before you leave the Metadata tab, click Save to save the updated metadata.

8 Click on the File List tab to set up new SSAP components.

- The File List Tab displays files in the local directory to the left and files in the selected SSAP component to the right. On the bottom left is a directory listing and a method to move through the directory tree on the local machine. Delete and Reset buttons -- both active – are to the right.

9 Click on the File Type button to select the SSAP component to manipulate.

10 Choose one of the menu items.

11 Select a file (or files) from the left window to add to the component.

12 Click the Add Arrow button to add the files. They will appear in the right window because they are now part of that SSAP Component.

13 Click Main to get back to the Main tab.

14 On the Main tab, click Submit to send the new SSAP to Data Server. When finished, a message should pop up that says “SSAP Successfully inserted to the Data Server”. The SSAP has been updated at the Data Server.

26.12.13 Placing the Science Software Executable Package (SSEP) on the Data Server

In order to be able to run a PGE within the ECS system, the EXE TAR file has to be inserted to the Science Data Server. This tar file consists of all files needed to run a PGE, except for input data files. This includes the executables, any scripts, and the SDP Toolkit message files.

26.12.13.1 Assembling a Science Software Executable Package (SSEP)

This section describes how to assemble a Science Software executables Package (SSEP) and create a corresponding Target MCF. An SSEP is a UNIX tar file that contains PGE executables and SDP Toolkit message files.

In order to Insert a PGEEXE tar file into the Science Data Server, a corresponding Target MCF (.met) must be generated before insertion. Such an ASCII metadata ODL file can be obtained by editing an existing template ODL file with the information of the specific PGE. The following procedures describe how to assemble a PGEEXE tar file and create an ASCII metadata ODL file.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. PGE executables and message files required by this PGE are available to make a SSEP.

To create an SSEP, execute the steps that follow:

1 At the UNIX prompt on an AIT Sun, type **mkdir *SSEPpathname***, press **Return**.

- The ***SSEPpathname*** is the full path name of a *new* directory that will contain all the files to be placed into the SSEP as well as the SSEP itself.
- It is recommended that ***SSEPpathame*** be named with a convention that indicates the PGE for which a SSEP will be created. For example, type **mkdir PGE35.ssep**, press **Return**.

2 At the UNIX prompt on the AIT Sun, type **cd *SSEPpathname***, press **Return**.

- The ***SSEPpathname*** is the directory name of the new directory created in step 1.

3 At the UNIX prompt on the AIT Sun, type **cp *pathname/file1 pathname/file2 … pathname/filen* .**, press **Return** (note the "dot" and then space at the end of the command).

- The ***pathname/file1, pathname/file2,…pathname/filen*** represents a list of path names and file names (delimited by spaces) to copy into the current directory, ***SSEPpathame*** (the "dot" represents the current directory and must be last in the command).
- For example, type **cp /data/MODIS/pge/PGE35.exe /data/MODIS/mcf/mod35.mcf /data/MODIS/MOD_13453 .**, press **Return** (note the space and then "dot" at the end of the command).
- The files copied into this directory should be the PGE executable, any shell scripts or other executables that are part of the PGE and SDP Toolkit message files.
- Files can be individually copied into the ***SSEPpathame*** directory. For example, type **cp /data/MODIS/pge/PGE35.exe .**, press **Return** (note the space and then "dot" at the end of the command). Repeat for each file needed in the SSEP for this PGE.

4 At the UNIX prompt on the AIT Sun, type **tar cvf *SSEPfilename*.tar ***, press **Return**.

- The ***SSEPfilename*.tar** is the file name for the SSEP tar file. The file name extension .tar is recommended but not required.
- The asterisk (*****) is a file name wildcard that represents all files in the current directory. This will place all files in the SSEP tar file.
- Once created, the contents of the SSEP tar file can be viewed by typing **tar tvf *SSEPfilename*.tar**, press **Return**.
- Do not apply compression (*e.g.* UNIX compress or gzip) to the tar file.

5 At the UNIX prompt on the AIT Sun, type **cp *filename.met.tpl filename.met*,** press **Return**.

- The ***filename.met.tpl*** is the file name of the template Target MCF for this SSEP. If a template is not available, see Appendix D or use one used for another SSEP.
- The ***filename.met*** is the file name of the Target MCF to be tailored for this SSEP.

6 At the UNIX prompt on the AIT Sun, type **vi *filename.met***, press **Return**.

- The ***filename.met*** is the Target MCF for this SSEP.
- This command invokes the *vi* editor. Edit the ***filename*.met** with the specific information for the SSEP to be inserted.
- The following guidelines should be followed when editing on the Target MCF (***filename*.met**):
  - The value for the VERSIONID object should be filled out with the proper PGE version. For example: “1” .
  - In the INFORMATIONCONTENTCONTAINER object,

    The value for the PARAMETERNAME object of the class “1” should be filled out with the PGE name. For example: “BTS”.

    The value for the PARAMETERNAME object of the class “2” should be filled out with the PGE Science Software Version. For example: “1”.

    The value for the PARAMETERNAME object of the class “3” should be filled out with the Platform Name. For example: “IRIX”.

    The value for the PARAMETERNAME object of the class “4” should be filled out with the Platform Version. For example: “6.5”.

    The value for the PARAMETERNAME object of the class “5” should be filled out with the date to perform the Insertion. For example: “970319”.

    The value for the PARAMETERNAME object of the class “6” should be filled out with the time to perform the Insertion. For example: “14:45:00”.

7 Save the changes made to the SSEP’s Target MCF (***filename*.met**) and exit the editor.

- The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, press **Return**.

- For other editors, refer to that editor’s documentation

SSEP procedures continued:

1 At the UNIX prompt on the AIT Sun, type **mkdir *SSEPpathname*** then press the **Enter** key. For example, type **mkdir MOD35.ssep, press Enter.**

- The ***SSEPpathname*** is the full path name of a *new* directory that will contain all the files to be placed into the SSEP as well as the SSEP itself.

2 At the UNIX prompt on the AIT Sun, type **cd *SSEPpathname***, then press the **Enter** key.

- The ***SSEPpathname*** is the directory name of the new directory created in step 1.

3 At the UNIX prompt on the AIT Sun, type **cp *pathname/file1 pathname/file2 … pathname/filen* .**, then press **Enter** (note the space then the “dot” at the end of the command).

- The ***pathname*** is the location of the files. The ***file1, file2,…filen*** represents a list of file names (delimited by spaces) to copy into the current directory, ***SSEPpathame*** (the “dot” represents the current directory and must be last in the command). For example, type **cd /data/MODIS/pge/MOD35.pge /data/MODIS/mcf/MOD35.mcf /data/MODIS/MOD_13453 ., press Enter.** (note the space then the “dot” at the end of the command).
- For the synthetic PGE, only the executable needs to be copied.

4 At the UNIX prompt on the AIT Sun, type **tar cvf *SSEPfilename*.tar ***, then press the **Enter** key.

- The ***SSEPfilename*.tar** is the file name for the SSEP tar file.
- The file name extension .tar is recommended but not required.
- The asterisk (*****) is a file name wildcard that represents all files in the current directory, which will place all files in the SSEP tar file.
- Once created, the contents of the SSEP tar file can be viewed by typing **tar tvf *SSEPfilename*.tar**, then press the **Enter** key.

5 At the UNIX prompt on the AIT Sun, type **cp *filename.met.tpl filename.met*,** then press the **Enter** key.

- The ***filename.met.tpl*** is the file name of the Target MCF for this SSEP.
- For the synthetic PGE, the **met** file has already been renamed and modified for use by the student when the file was unpacked.

6 At the UNIX prompt on the AIT Sun, type **vi *filename.met***, then press the **Enter** key.

- The ***filename.met.tpl*** is the Target MCF for this SSEP.

7 Edit the ***filename*.met** with the specific information for the SSEP to be inserted.

- The value for the **VERSIONID** object should be filled out with the proper PGE version.
- In the **INFORMATIONCONTENTCONTAINER** object enter the following:
- The value for the **PARAMETERNAME** object of the **class "1"** should be filled out with the PGE name. The synthetic PGE should be **"userid".**
- The value for the **PARAMETERNAME** object of the **class "2"** should be filled out with the PGE Science Software Version.
- The value for the **PARAMETERNAME** object of the **class "3"** should be filled out with the Platform Name.
- The value for the **PARAMETERNAME** object of the **class "4"** should be filled out with the Platform Version.
- The value for the **PARAMETERNAME** object of the **class "5"** should be filled out with the date to perform the Insertion.
- The value for the **PARAMETERNAME** object of the **class "6"** should be filled out with the time to perform the Insertion.

8 Save the changes made to the SSEP's Target MCF (***filename*.met**) and exit the editor.

- The specifics depend upon which editor is being used. If using *vi*, the command sequence to enter is **:wq**, then press the **Enter** key.

26.12.13.2 Insert a Science Software Exec Package (SSEP) onto Data Server

Science software, like any other data that are managed in the ECS, must be placed on the Science Data Server. A program called the Insert EXE TAR Tool can be used for Inserting a Science Software Executable Package into the Data Server.

Detailed procedures for tasks performed by the SSI&T operator are provided in the sections that follow.

Assumptions:

1. The ESDT called PGEEXE has been installed on the Science Data Server.
2. A Target MCF (.met) for this PGEEXE tar file has been created for the Insert.
3. The PGEEXE tar file has been created .

The following Servers/Services must be up and operational:

Science Data Server, Storage Management.

To Insert the SSEP to the Science Data Server, execute the steps that follow:

1 From the SSIT Manager, click on the **Tools** menu, then choose **Data Server** and then **Insert EXE TAR**.

- An xterm with title "SSIT: PGE Executable Tar File Insertion" will be displayed.

2 At the program prompt **Configuration filename? (enter for default: ../../EcDpAtInsertExeTarFile.CFG)**, press **Return**.

3 At the program prompt **ECS mode of operations?**, Type **<*mode*>** press **Return.**

- **<*mode*>** can either be **OPS** or **TS1**.

4 At the program prompt **Name of PGE?**, type ***PGEName***, press **Return**.

- The ***PGEName*** is the name of the PGE for which this static granule is being Inserted. For example, type **PGE01**, press **Return**.
- The ***PGEName*** must match exactly the PGE name entered into the PDPS for this PGE.

5 At the program prompt **Science software version of PGE?**, type ***SSWversion***, press **Return**.

- The ***SSWversion*** is the version of the science software that is being Inserted in this SSEP. Press **Return** to accept the default or enter in a version and press **Return**.

6 At the program prompt **Staged filename to insert (including Full path)?**, type ***pathname*/*SSEPFileName***, press **Return**

- The ***pathname*/*SSEPFileName*** is the full path name and file name of the SSEP tar file to be Inserted. For example, type **/data/MOD35/ssep/PGE35_1.tar**, press **Return**.
- The SSEP tar file must not be compressed (*e.g.* with UNIX compress or gzip).

7 At the program prompt **Associated ASCII metadata filename to insert (including Full Path)? *pathname/SSEPFileName.met*)?**, press **Return**.

- The default is the file name of the granule to insert with the .met file name extension. If the default is not correct, then the file name of this file must be entered.

8 At the program prompt **Top level shell filename within tar file?**, type ***ExecFileName***, press **Return**.

- The ***ExecFileName*** is the file name of the top level executable or script within the SSEP tar file. It should be the same as was entered into the PDPS/SSIT Database Update GUI.
- The SSEP will be Inserted to the Science Data Server.

9 At the program prompt **Hit return to run again, ‘q <return>’ to quit:** type **q** and press **Return** to quit or just press **Return** to insert additional dynamic granules.

- If continuing, repeat steps 3 through 8.

Example of a successful insertion of a SSEP EXE TAR:

PGE Executable Tar File Insertion Script
Configuration filename? (enter for default:../../cfg/EcDpAtInsertExeTarFile.CFG)
ECS Mode of operations? (enter for default: OPS)
OPS
Name of PGE? (enter for default: PGE07)
PGE07
Science software version of PGE? (enter for default: 0)
0

Staged filename to insert (including FULL path)? (enter for default: **/home/emcleod/SSEP/MODPGE07.tar**)
Associated ASCII metadata filename to insert (including FULL path)? (enter for default: **/home/emcleod/SSEP/MOD_PR10.tar.met**)
Top level shell filename within tar file? (enter for default: **/home/emcleod/SSEP/MOD_PR10.exe**)
MOD_PR10.exe (note: this entry is done a second time) Note: If you get **core dump**, execute using “dbx command: type in: **dbx filename .exe**. This will help isolate error message that caused core dump.
Warning: Could not open message catalog "oodce.cat"
EcDpAtInsertExeTarFile: Process Framework: ConfigFile
../../cfg/EcDpAtInsertExeTarFile.CFG ecs_mode OPS
Performing INSERT......
incomplete group entries in the configfile, using default G1
Trying to make a request to [MDC:DSSDSRV]
incomplete group entries in the configfile, using default
Trying to make a request to [MDC:DSSDSRV]
incomplete group entries in the configfile, using default
Insert to Data Server and PDPS database update successful for:
  PGE name = 'PGE07'
  Ssw version = '0'
  ESDT = 'PGEEXE'
  ESDT Version = '0'
  staged file = '/home/emcleod/SSEP/MODPGE07.tar'
  metadata file = '/home/emcleod/SSEP/MOD_PR10.tar.met'
  Top level shell name = 'MOD_PR10.exe'
Inserted at UR:
'UR:10:DsShESDTUR:UR:15:DsShSciServerUR:13:[MDC:DSSDSRV]:14:LM:PGEEXE:1787'
Hit return to run again, 'q <return>' to quit:

26.13 PGE Planning, Processing and Product Retrieval

The procedures for creating data products using the production planning and processing tools are described in detail in Chapter 13, Production Planning and Chapter 14, Production Processing. Please refer to those chapters of this document for additional information.

26.14 Postprocessing and General Investigation

An important part of SSI&T is verifying that the output files produced at the DAAC are identical (within particular tolerances) to the test output files delivered with the DAPs. A successful comparison is a strong indication that the porting of the science software from the development facility at the SCF to the operational facility at the DAAC has not introduced any errors.

A number of file comparison tools are available during SSI&T via the SSIT Manager GUI or they can be invoked from the UNIX command line. Two tools are available for comparing HDF or HDF-EOS files, one tool for comparing ASCII files, and another tool for assisting in comparing binary files.

It is assumed that the Instrument Team has delivered test output files (produced at their SCF) with which to perform the comparison.

26.14.1 Examining PGE Log Files

Three log files are produced by PGEs during runtime: the Status log, User Log, and the Report log. These log files are written by the SDP Toolkit and by the science software using the Toolkit's Status Message Facility (SMF). The location of these log files is specified in the Process Control File (PCF). When the PGE is built and run with the SCF version of the SDP Toolkit, the location and file names of the log files can be set as desired. When the PGE is built with the DAAC version of the SDP Toolkit and run within the PDPS, the location and file names of the log files is set by the system in the instantiated PCF.

The Status log file captures all error and status information. The User log file captures a subset of messages that are more informational. The Report log file captures arbitrary message strings sent by the PGE.

The section aforementioned describes how to examine log files produced by PGEs that have been built with the SCF version of the SDP Toolkit and run from the command line.

The section aforementioned describes how to examine log files (within the Production History) produced by PGEs that have been built with the DAAC version of the SDP Toolkit and run within the PDPS.

26.14.2 ECS Mechanisms for Capturing Information about a PGE's Execution

Note: This information was provided by ECS expressly for this document. It describes two tar files that are generated by the ECS Planning and Data Processing System (PDPS). One of these file types is generated for each execution of a PGE that is run within the ECS environment. The particular file type that is generated depends on the success or failure of the PGE execution. The ECSDataGranule metadata contains a reference to the Production History.

26.14.2.1 ECS Production History (PH)

For each successful execution of an instance of a PGE (represented by a data processing request - DPR), the processing history is captured in the production history tar file. The production history tar file is generated and archived by ECS (inserted into the Data Server) upon PGE completion. The PH is a UNIX tar file. The PH contains multiple files. Each PH can be uniquely retrieved from the Data Server. A science user has the option to acquire the PH when ordering a science granule. The name of the tar file will include a UR for the PH.

The tar file can be untarred and contains numerous component files. After untarring the PH, the file names of the components provide the traceability to a particular DPR (data processing request or job) that ran under ECS. Please note that the dprid is a concatenation of an abbreviated form of the PGE name (e.g., for MODIS PGE01, it would be MoPGE01#version) with a date/time stamp that is the start of the processing time for the DPR. The dprid string may have some trailing 0’s filled in.

A PHcomponentFile can be examined to identify the components within the PH.

Other component files are:

- The PH log file or processing log: Named PGEname#versionMMDDhhmm.Log – contains the DPR ID, the actual command used to run the PGE (.in), resource usage information, and the PGE exit status. It also contains a listing of output products generated, their file paths and file sizes.
- The PCF file: Named PGEname#versionMMDDhhmm.Pcf – contains the actual instantiated PCF used when running the instance of the PGE (this DPR). Note that the PCF contains URs for all inputs to this execution of the PGE. There is one UR for each input granule. If a granule is a multi-file granule, the same UR will appear (repeat itself) for each inputs file of the granule.
- The Production Log file: Named PGEname#versionMMDDhhmm.ProdLog – Contains the DPR ID, the PGEID, and resource usage information (same as in the .Log file). Resource usage information includes:
    - CPU time in application
    - CPU time in system
    - Physical memory
    - Max. resident set size
    - Avg. shared text size
    - Avg. shared data size
    - Avg. shared stack size
    - Page reclaims
    - Page faults
    - Swaps
    - Block input ops

- Block output ops
- Messages sent
- Messages received
- Signals received
- Voluntary context switches
- Involuntary context switches

- The Profile file: Named PGEname#versionMMDDhhmm.Profile – Contains the environment variables defined during the execution of the PGE including the contents of the PATH environment variable.
- The SDP Toolkit log files:
  - The Report log file: Named PGEname#versionMMDDhhmm.TkReport – This is the same log file that the SDP Toolkit generates when the PGE runs outside of ECS. (See SDP Toolkit documentation.)
  - The Status log file: Named PGEname#versionMMDDhhmm.TkStatus – This is the same status log file that the SDP Toolkit generates when the PGE runs outside of ECS (See SDP Toolkit documentation.)
  - The User log file: Named PGEname#versionMMDDhhmm.TkUser – This is the same user log file that the SDP Toolkit generates when the PGE runs outside of ECS (See SDP Toolkit documentation.)

26.14.2.2 ECS Failed PGE Tar File

For each unsuccessful execution of an instance of a PGE (represented by a data processing request or DPR), a failed PGE tar file (may also be called the History Log or HL) is created as a diagnostic tool for the science software provider/analyst. The Failed PGE tar file is generated and archived by ECS (inserted into the Data Server) upon the abnormal completion of the PGE execution. Similar to the PH, the Failed PGE tar file is a UNIX tar file. An Instrument Team user would acquire a FailedPGE tar file by interacting with the DAAC operations staff. The name of the tar file will include a UR for the FailedPGE tar file.

The Failed PGE tar file contains numerous component files that can be examined after untarring. After untarring the FailedPGE tar file, the PHcomponentFile can be examined to identify the components within. The component files are the same as those in the PH. In addition, if the PGE's execution ended with a core dump, the core file is included in the tar file. Since the FailedPGE tar file results from a failed execution of a PGE, the contents will reflect the point of the failure.

Operationally, the ECS SDSRV error logs are examined to view the PH or Failed PGE tar files that were generated.

The error log entries referencing a PH tar file are of the format:

> **:PH.version:dbid:1.BINARY** where version is the ESDT version number for the Production History ESDT, and dbid is the database ID, a unique identifier within SDSRV.

The error log entries referencing a FailedPGE tar file are of the format:

> **:LM: FAILEDPGE.version:dbid:1.BINARY** where version is the ESDT version no. for the FailedPGE ESDT, and dbid is the database ID, a unique identifier within SDSRV

Operations staff may use the dbid to access the corresponding tar file itself from the data server. The file may then be untarred.

26.15 Examining PDPS-Related Scripts and Message Files

This section describes how users may access files, in addition to the PGE-produced log files, which are created during the execution of a DPR job and which may hold information useful in tracing processing problems.

Some of these files are written by default to directory paths that can only be accessed on either the SGI processor machine or one of the Sun workstations. More detailed descriptions of these files and the conditions under which they are generated will be supplied in future Green Book versions.

26.15.1 Examining AutoSys JIL Scripts

JILxxxxxxxxx is the Job Information Language (JIL) script that defines the DPR job to **AutoSys** and which must be submitted to the **AutoSys** Database before a DPR job can be run. The name of the file created is system-generated and begins with the characters 'JIL' followed by nine characters (e.g. JILAAAa0066c).

Sample file content:

```
insert_job: 5251_823122483_1
job_type: command
command:  /usr/ecs/{mode}/CUSTOM/data/bin/sgi/EcDpAtExecutionMain
5251_823122483_1
machine: spr1sgigsfc
std_out_file: /home/cboettch/mockpge_msfc/out/dpat_std.out
std_err_file: /home/cboettch/mockpge_msfc/out/dpat_std.err
profile: /usr/ecs/<MODE>/CUSTOM/data/bin/sgi/EcDpAtRunProfile.sh
```

To examine JILxxxxxxxxx scripts on the AIT Sun, execute the procedure steps that follow:

1 At the UNIX prompt on an AIT Sun, type **cd** ***JILscriptPathname***, press **Return**.
- The ***JILscriptPathname*** is the full path name to the location of the JILxxxxxxxx scripts to be examined.

2 At the UNIX prompt on the AIT Sun, type **vi** ***JILscriptFilename***, press **Return**.
- The ***JILscriptFilename*** is the file name of the JILxxxxxxxx script to be examined.
- This brings up the file named ***JILscriptFilename*** in the *vi* editor.
- Any text editor may be used such as *emacs*. For example, **emacs** ***JILscriptFilename***, press **Return**.

26.15.2 Examining Application Log Files (ALOG)

Most of the custom code used during SSI&T routinely produces log files. For example, the SSIT Manager produces a log file named **EcDpAtMgr.log** and the tool used to Insert SSEPs to the Data Server (EcDpAtInsertExeTarFile.sh) produces a log file named **EcDpAtInsertExeTarFile.log.** These files are placed in the directory in which the tool was executed. If the **SSIT Manager** is run from the user's home directory, then the log files for each of the associated tools will be found in the user's home directory. Log files are produced at the first invocation of the tools, even if no messages are written to them. During subsequent use of the tools, the associated log files will be appended.

Log files are generally named according to the convention:

***ApplicationName*.log**

where ***ApplicationName*** is replaced with the name of the tool's executable binary. For tools that are shell scripts (*e.g.* .sh files), the shell name is left out of the log file name. For example, the tool EcDpAtInsertStaticFile.sh produces a log file named **EcDpAtInsertStaticFile.log** and not EcDpAtInsertStaticFile.sh.log.

Where an **SSIT Manager** application has been run using login **cmts1**, pw: **ecsu$er**, the log files will be found using path: **/usr/ecs/{MODE}/CUSTOM/logs/.**

Connectivity failures have been encountered when installing **ESDTs , MCFs** and **.met files**. The term bounce the servers has been widely used in conjunction with the effort to re-install or delete files. **Bounce** means to **shut down a server** and then **bring them back up** to rid the servers of unwanted or old bindings. The nature of what needs to be done is outlined as follows:

1 Install or Delete **ESDTs -** the **SDSRV** needs to be bounced after installation or removal of **ESDT**s to allow for a refresh of the **Connectivity** cell management.

2 For **PGE…..odl, MCFs** and **.met files,** bouncing the server **SDSRV** needs to done after installation and reinstallation.

3 This can be done by logging into **ECS Assistant** for each server. The login should be with generic **ID**: and **PW**:**,** and then press **Enter** Key**.**.

26.16 PDPS Troubleshooting - the PGE Job Has Failed

The procedures for troubleshooting problems with PGEs that have run are described in detail in Chapter 14, Production Processing. Please refer to that chapter of this document for additional information.

26.17 DPREP

DPREP (data preprocessing) consists of sets of PGEs that use a statistical approach to convert Level 0 (L0) ephemeris and attitude ancillary data for a particular EOS satellite (i.e., Terra, Aqua, or Aura) into SDP Toolkit native binary format without altering or modifying the scientific content of the granules.

DPREP PGEs are supplied by ECS, unlike most PGEs, which are provided by the Science Computing Facilities that ECS supports. DPREP supports Terra and Aqua operations and will eventually support Aura operations.

For detailed information on DPREP refer the most recent version of the following EMD project documents:

- 500-EMD-001, Terra Spacecraft Ephemeris and Attitude Data Preprocessing.
- 500-EMD-002, Aqua Spacecraft Ephemeris and Attitude Data Preprocessing.
- 500-EMD-003, Aura Spacecraft Ephemeris and Attitude Data Preprocessing.

26.18 PGE Chaining

26.18.1 Chaining PGEs

1 Create PRs (so that DPRs) for the PGEs to be chained.

- This can be done using PR Editor. Follow the same procedure as creating independent PR.
- A few points need to be noticed. Let's say among the chained PGEs, the output of PGE A will be the input of PGE B.1) In ESDT odl for this shared granule, "DYNAMIC_FLAG" has to be set to "I", i.e., dynamic internal. 2) First create PR for PGE A, then for PGE B. Otherwise PGE B PR may not be able to be generated.

2 Create the plan for a bunch of PRs that are chained.

- In Work Bench GUI, 1) pull down "file" menu and select "new" to create the new plan; 2) highlight all PRs that are chain by clicking on their names on "unscheduled" area of Production Request area; 3) click schedule button to schedule these PRs.

3 Activate the plan.

- In the Workbench GUI, click "activate" button, a GUI will pop up to ask for saving the plan. Answer "yes". Then another GUI will pop up to confirm whether to really activate the plan. Answer "yes" and the lowest level of DPR(s) in the chain will kick off.
- In the PDPS database, the PlDataProcessingRequest table is where the PRs are successfully generated, the "completionState" for all DPRs in the chain are "NULL". When the plan is successfully activated, the "CompletionState" for lowest level of DPR(s) is changed from "NULL" to "STARTED". The high level of DPR(s) in the chain is changed from "NULL" to "CQ_HOLD". Eventually, the low level of DPR(s) finish so that the input for high level of DPR(s) become available, Then the high level DPR(s) kick off and the "CompletionState" then changes from "CQ_HOLD" to "STARTED".

26.19 Updating the Orbit Model

26.19.1 Introduction to Updating the Orbit Model

To determine real time the latest Orbit Start times, Orbit Period, Path Number and Orbit Number, PDPS takes in specific information about the orbit of the satellite during initial SSI&T. This information then becomes the basis for predictions of future orbit start times and numbers. Because this value is accurate within a fraction of a second of time, the satellite may "drift" or a correction to orbit, known as a "burn" may have been applied. Therefore, the satellite Orbit Start Time can get out of sync either +/- with reality. The consequences are an elapse in time that will affect the Production Request Editor's ability to find a granule that should match with a DPR, or an incorrect Orbit Time could be passed to the PGE. The update of Orbit parameters will be done weekly at a specific time with scripts specifically written to extract the new Orbit Parameters from the most recent DPREP output file. These parameters will be inserted manually to the ORBIT.ODL file and then SSI&T personnel will re-register the Orbit.ODL file in the PDPS. The DAAC Support Help Desk Team is responsible for knowing when changes to Orbit location have taken place from the Flight Dynamics Systems (FDS). The DAAC Support Help Desk will maintain a Knowledge Base with backup procedures for contingencies concerning Orbit Model updates. DPREP processing will be the most likely place to experience a failure due to Orbit time sync error encounters. The restoration of Orbit parameters with new values from FDS will most likely be necessary. The following procedures are provided to bring about an updated Orbit Model within ECS.

26.19.2 Procedures to Update the Orbit Model

Upon receipt of updated orbit parameters: ORBIT_ NUMBER,
ORBIT_PERIOD,
ORBIT Path Number and
ORBIT_START Time.

Proceed with the following steps.

1 Telnet or Rlogin to location (ais) system where ODL files are stored. i.e., **"/usr/ecs/OPS/CUSTOM/data/DPS/ODL"**

2 Select the ORBIT.odl that is currently being used.

3 Using vi, update the following files with the new parameter values received:
- **ORBIT_AM1.odl** and/or **ORBIT_EOSAM1.odl** if they both are in use.

4 Have someone double check your entries for accuracy before proceeding to the SSIT Manager for registering the new ODL file in the PDPS system.

5 For **Test Data** only; determine the Instrument PGE ODL that will be updated. MISR, MODIS etc.

- Using **vi, update the corresponding PATHMAP_Instrument_.odl** file with the new parameter values received.
- Ensure that the ABSOLUTE_PATH and MAPPED_PATH parameters agree with those in the new ORBIT_XXXX.odl.

6 SSI&T personnel will execute an Orbit Model Update by running a Dummy PGE established for this purpose at each of the DAACs. **Note:** A dummy PGE is run since a normal PGE cannot be re-registered if any DPRs exist in the system.

7 Notify DAAC Operations Supervisor that the Orbit Model has been updated. He will make a log entry of such action taken and may request the old computed values and the new replacement values be provided. The Supervisor will ensure that the orbital change is within several seconds, the expected change and not minutes!

26.20 Troubleshooting and General Investigation

This section is primarily meant to serve as a reference to SSIT personnel for use in diagnosing problems encountered during testing of PGEs in the ECS system. It is divided into subsections by category of failure. Some types of failures are common to many steps in the SSIT process, so their investigation follows a common path. Thus, the SSIT user should refer to the following sections on the basis of the underlying process (e.g., file insertion to the Science Data Server) rather than by the distinct stage in the SSIT procedure.

This section is divided into six main categories of failure, each section will discuss ESDT Installation failure

- File Insert Failure
- File Acquire Failure
- DPR Generation Failure
- Scheduling a DPR Failure
- PGE Execution Failure

26.20.1 Description

ESDT versioning has been implemented in this release, which means a DLL can be shared by more than one ESDT descriptor file. In this release, the DLL file that will be used during installation is specified in the ESDT descriptor file. Therefore during ESDT installation, a user only need to provide ESDT descriptor file name instead of ESDT descriptor file and DLL file.

26.20.2 Handling an ESDT Installation Failure

During ESDT installation, there are two types of failure. One is general failure, which means no ESDT can be installed. This case is usually caused by improper operations, which means the Servers may not be brought up properly, or a login is expired etc. The other is particular ESDT installation failure, which means certain type of ESDT can not be installed, but other ESDTs may be able to installed. This case is relatively hard to for a user trace the problem. The problem is usually caused by particular attributes in the ESDT descriptor file, which means the ESDT descriptor file is incompatible with Science Data Server Database table definitions and also some attributes in the ESDT descriptor file may violate the validation rule defined in the Data Dictionary. Therefore as a general rule, a user needs to identify the failure type first. The following sections are general guidance for a user to handling an ESDT installation failure.

Check the status of the servers: When you install an ESDT, there are certain servers that have to be brought up properly. If either or all the servers are down or hasn't been brought up properly, ESDT installation fails. Please refer to section 18.3 for more information about the servers, which are involved when installing an ESDT.

Check the ESDT descriptor file and the DLL file: Inside Science Data Server configuration file, which is EcDsScienceDataServer.CFG (at /usr/ecs/<mode>/CUSTOM/cfg), there are entries that indicate where to find ESDT descriptor file and DLL. Those entries are DSSDESCINPUTDIR and DSSDLLDIR. In the ESDT descriptor file, there is an attribute name that is DLLName. The DLL file should be resided at DSSDLLDIR. If the DLL file name indicated in ESDT descriptor file can not be found under the DSSDLLDIR, a failure of installation will occur.

Remove the ESDT descriptor file properly before installing an updated version of the same ESDT : You don't have to worry about removing the descriptor file before installing an ESDT, when you are installing a totally new ESDT. But when you install an updated version of an ESDT, which has already been installed, you need to remove it first. Please refer to section 18.2 for procedure of removing an ESDT. Sometimes ESDT doesn't get removed properly, and then if you try to install it, the installation fails.

Check the Science Data Server ALOG File: During ESDT installation the Science Data Server writes status messages to two log files, EcDsScienceDataServer.ALOG and EcDsScienceDataServerDebug.log. Entries to the ALOG file should include the ShortName of the ESDT. If the ShortName does not appear with a time stamp that reflects the time of the attempted installation, then the request of installation was not communicated to SDSRV. This might be the case if the SDSRV subsystem is not running, These logs files can be inspected further for other errors, if the ESDT ShortName does appear.

26.20.3 Insert File Failure

Files may be inserted into the Science Data Server in a variety of ways, depending on the type of file. In general, files must be accompanied by a descriptive metadata (.met) file in order for the Science Data Server to process them successfully. The types of files to be inserted include DAP (Delivered Algorithm Package) files, input and output science data granules, and PGE executable tar files (PGEEXE.tar). The Science Data Server provides a test driver to insert a file to SDSRV. The SSIT Manager GUI also provides various tools to insert a file, depending on its type. Also, one phase (Destaging) of the execution of a PGE by the system involves the insertion of output data products to the Science Data Server.

For inserting files using the SDSRV test driver or the SSIT Manager, the user first needs to prepare a metadata file to go with the file to insert into Science Data Server. Output file insertions during destaging use metadata files generated by the PGE.

Files that are to be inserted to the Science Data Server using the SSIT Manager, in general, must be made known to the PDPS database in advance. This includes dynamic and static data files, and PGE executable tar file. This prerequisite is fulfilled by registering a PGE with the PDPS database. Once a data type has been made known to the PDPS database via PGE registration, files of that type may be inserted to the Science Data Server regardless of the PGE for which they will serve as input or output. The PGEEXE.tar file can only successfully insert following registration of the particular PGE.

26.20.3.1 Handling an Insert Failure

If the Science Data Server returns a message indicating that the insertion has failed, or in the case of a Destaging failure, a number of things can be done to diagnose the problem causing the failure.

Check for ESDT in SDSRV Database: Check the Science Data Server for the ESDT corresponding to the file type to be inserted. This is done in either of two ways. The first is to enter a few SQL commands on the UNIX command line. The second utilizes a database viewer GUI. With either method, the aim is to verify that the ESDT required is listed in the Science Data Server data base in a table called DsMdCollections.

Table 26.20.3.1-1 lists the steps required to view the SDSRV data base using SQL commands.

Table 26.20.3.1-1. Viewing the SDSRV Database Using SQL Commands - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	Login to AIT computer which supports SDSRV using a generic SSIT account.	
2	isql –U <sdsrvusername> –P<password> -S<sdsrvserver>	press Return
3	use <databasename>	press Return
4	Go	press Return
5	Select ShortName from DsMdCollections where ShortName = <ESDTshortname>	press Return
6	Go	press Return

Table 26.20.3.1-2 lists the steps required to view the SDSRV data base using a database viewer.

Table 26.20.3.1-2. Viewing the SDSRV Database Using the Database Viewer GUI - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	login to AIT computer which supports SDSRV using a generic SSIT account.	
2	(if necessary) setenv DISPLAY <machinename>:0.0	press Return
3	Wisqlite <MODE> &	press Return
4	enter into GUI login: DBUSERNAME: <sdsrvusername> DBPASSWD: <password>, DSQUERY: <sdsrvserver>	
5	select database by clicking DATABASE button	
6	type command as in 5 of above table	Click “execute”

For both methods, the aim is to locate the ESDT corresponding to the file to be installed. In the DsMdCollections table, the ESDT is located by the shortname. If the shortname is not listed, then the ESDT must be inserted to the Science Data Server before the file insertion could succeed..

26.20.4 Acquire Failure

Files are acquired from the Science Data Server either through the SSIT Manager, DAP acquire tool (it can be used for any type of file acquire), or using a test driver from SDSRV. Additionally, files are acquired by the system during the setup and execution of a PGE. Failure of an acquire is similar to insertion failure, and the methods to diagnose and resolve the failure also resemble those for insertions.

26.20.4.1 Handling an Acquire Failure

Diagnosing an acquire failure involves inspecting various system log files and checking in directories involved with the process.

Check Science Data Server Log Files: The EcDsScienceDataServer.ALOG file should contain entries regarding the acquire activity and identify the file to be acquired by its ShortName. If the ShortName does not appear in the ALOG file, with a timestamp corresponding to the time of the attempted acquire, then SDSRV may not be running, or may not be communicating with other servers. If the ALOG file does contain entries for that ShortName, and indicates that two files (the file and its associated metadata file) are being distributed, the message in the ALOG file looks like following:

Msg: File 1 to be distributed: :SC:MOD03.001:1369:1.HDF-EOS
Priority: 0 Time : 07/29/98 12:35:42
PID : 24279:MsgLink :1684108385 meaningfulname
:DsSrWorkingCollectionDistributeOneDistributFile
Msg: File 2 to be distributed: SCMOD03.0011369.met

This indicates that SDSRV has completed its role in the acquire. Therefore the acquire script usually will indicate success.

Table 26.20.4-1. Viewing the EcDsScienceDataServer.ALOG file - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	login to AIT computer that supports the SDSRV subsystem using a generic SSIT account.	
2	cd /usr/ecs/mode/CUSTOM/logs	press Enter
3	vi EcDsScienceDataServer.ALOG	press Enter

If the ALOG contains the ShortName, and also contains an error showing that the data file time stamp does not match the time stamp required by the acquire, then the data file needs to be removed from the Science Data Server and reinserted. This is usually done using a script called DsDbCleanGranules.

The procedures for removing a data granule from Science Data Server are as follows:

1 In directory **/usr/ecs/<MODE>/CUSTOM/dbms/DSS/** of the SDSRV host, set appropriate environment variables for **DBUSERNAME**, **DBPASSWD**, **DSQUERY** and **DBNAME**.

2 Then type command **DsDbCleanSingleGranule SC:<ShortName.version>:<dbID>** (e.g. DsDbCleanSingleGranule SC:MOD000.001:1234) then press **Return**.

Check the Archive Server ALOG File: Acquire success from the Science Data Server is only part of the acquire process. Since any file entered into SDSRV is stored in the archive, the Archive Server must be involved during an acquire. Thus, it may be useful to inspect the Archive Server ALOG file (EcDsStArchiveServer.ALOG) to check for error messages associated with the ShortName of the file type.

Table 26.20.4-2. Viewing the EcDsStArchiveServer.ALOG file - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	login to AIT computer that supports the Archive Server subsystem using a generic SSIT account.	
2	cd /usr/ecs/*mode*/CUSTOM/logs	press Enter
3	vi EcDsStArchiveServer.ALOG	press Enter

Check Staging Disk: During an acquire, files are copied to a staging area as an intermediate step before distributing them to their destination. As part of diagnosing an acquire failure it is useful to check the staging area to ascertain whether the files have completed part of their journey. Both the file and a subdirectory containing metadata information should be written to the staging area.

Table 26.20.4-3. Viewing the Staging Area - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	login to AIT computer that supports the Archive Server subsystem using a generic SSIT account.	
2	cd /usr/ecs/*mode*/CUSTOM/drp/*archivehost*/data/staging/user#	press Enter
3	ls –lrt	press Enter

Check Staging Server ALOG: If the failure occurs in copying the files to the staging area, then the Staging log files (EcDsStStagingDiskServer.ALOG or EcDsStStagingMonitorServer.ALOG) may reveal the cause.

Table 26.20.4-4. Viewing the EcDsStagingServer.ALOG file - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	login to AIT computer that supports the Archive Server subsystem using a generic SSIT account.	
2	cd /usr/ecs/*mode*/CUSTOM/logs	press Enter
3	vi EcDsStStagingDiskServer.ALOG or EcDsStStagingMonitorServer.ALOG	press Enter

Check the Space Available in the Staging Area: Failure can also be caused by a lack of space in the staging area.

Table 26.20.4-5. Checking the Space Available in the Staging Area - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	login to AIT computer that supports the Archive Server subsystem using a generic SSIT account.	
2	cd /usr/ecs/*mode*/CUSTOM/drp/*archivehost*/data/staging/user#	press Enter
3	df -k **.**	press Enter

26.20.5 Failure During DPR Generation

The creation of a Data Processing Request is an essential part of the SSIT process. There are many reasons for DPR creation failure to occur. During DPR generation, the DPR generation executable will turn subscriptionFlag from zero to non-zero, which needs subscription Server up running and the executable will also query Science Data Server for input granules. The cause of failure to generate a DPR could come from following errors:

- Incorrect information in PGE_ODL and ESDT ODL files, which generally references to the PGE registration incorrectly.
- Not all the required Servers up running properly, which generally reference to the System problem.
- ESDTs required for the PGE were not properly installed.
- Database queries failure, which generally reference to Database errors.

26.20.5.1 Handling DPR Generation Failures

To find out why a DPR generation fails, a user needs to look for the Production Request Editor ALOG file, which is EcPlPREditor.ALOG resided at /usr/ecs/<mode>/CUSTOM/logs of PLS host. The ALOG file needs to be inspected for evidence of the source of a failure. In addition, that ALOG may indicate that the ALOG files for other subsystems, such as SDSRV, may contain entries describing the errors. Another useful resource for troubleshooting the failure is to look for PDPS database, all the PGE registration information is kept in different tables. Inside the table PlDataTypeMaster, there is a column called subscriptionFlag, during a DPR generation, the DPR executable will turn the subscriptionFlag for all the ESDTs needed for the PGE from zero to non-zero, if the Flag for the dataTypeId (corresponding to ESDT) did not turn to non-zero, that indicates subscription trouble. On the other hand, inside the table PlDataTypeMaster, there is a column called dataServUrString, during a DPR generation, the DPR executable will turn the dataServUrString for all the ESDTs needed for the PGE from NULL to UR value for Science Data Server. If the PGE contains static files, the UR for the static files have to been included in PlDataGranuleShort table before a DPR can be successfully generated. For dynamic granules, the UR values will become available in the PlDataGranuleShort table if the granules are available during the DPR generation, if the dynamic granules are not available during a DPR generation, it will not cause a failure to generate a DPR.

26.20.6 Failure Scheduling a DPR

Problems scheduling a PGE for execution in the system occur when a DPR, scheduled through the Planning Workbench, does not get passed to AutoSys.

26.20.6.1 Handling a DPR Scheduling failure

There is ALOG file called EcPlWb.ALOG resided at /usr/ecs/<mode>/CUSTOM/utilities of PLS host, which should be inspected first.

Check the PDPS Data Base: PlDataProcessingRequest: During scheduling, the PDPS data base is updated to reflect a change in the state of the DPR. In the PDPS data base the

PlDataProcessingRequest table will show a value of “NULL” in the completionState field if the DPR did not get passed to AutoSys.

The table below lists the steps required to view the PDPS data base using a database viewer.

Table 26.20.6-1. Viewing the PDPS Database Using the Database Viewer GUI - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	login to AIT computer that supports PDPS using a generic SSIT account.	
2	(if necessary) setenv DISPLAY *machinename*:0.0	press Return
3	/home/opscm/dbr/dbrowser-syb	press Return
4	enter into GUI login*: PDPSservername, PDPSusername, PDPSpassword*	
5	select pdps_*mode*	
6	select “sample data” under “view” menu	
7	select PlDataProcessingRequest	

The table below lists the steps required to view the PDPS data base using SQL commands.

Table 26.20.6-2. Viewing the PDPS Database Using SQL Commands - Quick-Step Procedures

Step	What to Enter or Select	Action to Take
1	login to AIT computer that supports PDPS using a generic SSIT account.	
2	isql -Updpsusername -Ppassword -Spdpsserver	press Return
3	use pdps_*mode*	press Return
4	go	press Return
5	select * from PlDataProcessingRequest where dprId = “*dprID*”	press Return
6	go	press Return

Check the PDPS Database: PlDataGranuleShort: Input data granules which are ready for use by a PGE running in the system will have entries with full URLs in the PDPS database table PlDataGranuleShort, under universalReference. Standalone PGE, those not running as part of a PGE chain, need to have the full URL entered. If the PGE is running as part of a PGE chain, then the input granules are products of preceding PGEs in the chain. If, in the ESDT odl files prepared for those input granules, the dynamic flag is set to “external” instead of “internal”, then the DPR will go into AutoSys as soon as the input granules become available. If the dynamic flag is set as “internal” then the DPR should go into AutoSys regardless of the availability of the input granules. PGE execution will commence, and the color of the display in JobScape within AutoSys will change, as soon as the input granules are available.

Inspection of the PDPS database follows the procedures outlined above, with PlDataGranuleShort as the table, and universalReference as the field. The individual granules can be identified by their data type (dataTypeId).

26.20.7 Failures During Execution

PGEs scheduled for execution in the system follow seven stages of processing. Each has its own types and causes of failure. It’s always better to put the next stage on hold while executing the previous. Once you execute one stage successfully with Exit Code 0, then take the hold off for the next. It’s easier to investigate if there are any failures. For example, when your PGE starts executing the first stage put second and third stage on hold. When your first stage executes successfully, then take the hold off from the second stage while third stage is still on hold. Again, if that goes though successfully then you can take the hold off from the third stage.

After execution of each stage, click on the stage and then click on Job Console button on the left side at the bottom to see the Exit Code of it. If it is 0 that indicates successful execution but if you get non-zero exit code that indicates failure.

26.20.7.1 Handling Failures During Execution

Resource Allocation: The first stage of PGE processing in AutoSys is Resource Allocation. If this fails, the ALOG file of the Data Processing host can be checked to see whether the PGEEXE.tar file was successfully acquired. If there is an acquire failure, further evaluation proceeds as outlined in Acquire Failure, above.

The log files DPR#.ALOG and DPR#.err are stored in /usr/ecs/<MODE>/CUSTOM/logs directory on PLS host, can be inspected to find out the cause of the failure.

To inspect the Data Processing ALOG file, execute the following steps:

1 Login to the AIT computer that supports Data Processing.

2 At the UNIX prompt, **type cd /usr/ecs/*mode*/CUSTOM/logs**, then press **Enter**.

3 At the UNIX prompt, type **vi *DPR#. ALOG or DPR#A.ALOG***, then press **Enter.**

Staging: The Staging step in processing involves acquiring files from SDSRV. Thus, it should be handled as in Acquire Failure, above.

To inspect the Data Processing ALOG file, execute the following steps:

1 Login to the AIT computer that supports Data Processing.

2 At the UNIX prompt, **type cd /usr/ecs/*mode*/CUSTOM/logs**, then press **Enter**.

3 At the UNIX prompt, type **vi *DPR#. ALOG or DPR#S.ALOG***, then press **Enter.**

Preprocessing: Preprocessing rarely completely fails. It may not generate the system PCF file correctly. Placing a “hold” on the PGE execution stage (through AutoSys, in the JobScape GUI, the Job Console button brings up the Job Console - PGE processing stages can be put on hold using the “On Hold” button.). While execution is on hold, the system PCF can be inspected to see whether it matches expectations.

To inspect the Data Processing ALOG file, execute the following steps:

1 Login to the AIT computer that supports Data Processing.

2 At the UNIX prompt, **type cd /usr/ecs/*mode*/CUSTOM/logs**, then press **Enter**.

3 At the UNIX prompt, type **vi *DPR#. ALOG or DPR#P.ALOG***, then press **Enter.**

To view the system PCF, execute the following steps:

1 Login to the AIT computer that supports Data Processing.

2 At the UNIX prompt, **type cd /usr/ecs/*mode*/CUSTOM/pdps/*hostname*/data/DpPrRm/*hostname*_disk/*pgeId*/*dprId*_ *hostname***, then press **Enter**.

3 At the UNIX prompt, type **vi pgeId.Pcf**, then press **Enter.**

PGE Execution: Failures during PGE execution can be investigated using the Toolkit LogStatus file, the system PCF file, and by running the PGE from the command line using the environment set in the PGE profile file and PGS_PC_INFO_FILE points to the system *.Pcf. All of these are found in the runtime directory.

To inspect these files, execute the following steps:

1 Login to the AIT computer that supports Data Processing.

2 At the UNIX prompt, **type cd /usr/ecs/*mode*/CUSTOM/pdps/*hostname*/data/DpPrRm/*hostname*_disk/*pgeId*/*dprId*_ *hostname***, then press **Enter**.

3 At the UNIX prompt, type **vi pgeId.Pcf** or **pgeId.TkStatus**, then press **Enter.**

Postprocessing: Postprocessing does not often fail, but it may show as a failure in AutoSys if the Execution stage has failed.

To inspect the Data Processing ALOG file, execute the following steps:

1 Login to the AIT computer that supports Data Processing.

2 At the UNIX prompt, **type cd /usr/ecs/*mode*/CUSTOM/logs**, then press **Enter**.

3 At the UNIX prompt, type **vi *DPR#. ALOG or DPR#p.ALOG***, then press **Enter.**

Destaging: Destaging involves the insertion of output data granules into SDSRV. Thus the section on Insertions of Files to the Science Data Server, should be consulted for this case. Volume group has to be set up for all the inputs as well as outputs of the PGE in order to execute this stage successfully. If either or all volume groups are missing then this stage fails.

To inspect the Data Processing ALOG file, execute the following steps:

1 Login to the AIT computer that supports Data Processing.

2 At the UNIX prompt, **type cd /usr/ecs/*mode*/CUSTOM/logs**, then press **Enter**.

3 At the UNIX prompt, type **vi *DPR#. ALOG or DPR#I.ALOG***, then press **Enter.**

Deallocation: Rarely are there cases of failure in Deallocation.

To inspect the Data Processing ALOG file, execute the following steps:

1 Login to the AIT computer that supports Data Processing.

2 At the UNIX prompt, **type cd /usr/ecs/*mode*/CUSTOM/logs**, then press **Enter**.

3 At the UNIX prompt, type **vi *DPR#. ALOG or DPR#D.ALOG***, then press **Enter.**

Rarely are there cases of failure in Deallocation. When it does occur, inspect the Data Processing ERR file following the same procedures described above.

26.21 Using IQ Software to Create Reports

26.21.1 Creating Reports Using IQ Software

ECS no longer plans to offer a Report Generator GUI. Consequently, DAAC operations personnel must use other means to generate various types of reports.

IQ (Intelligent Query) software is a set of commercial off-the-shelf (COTS) products that provides flexible access to the PDPS database from which data for reports can be retrieved. The cost of that flexibility is a somewhat complicated process for initially setting up reports. However, once a particular type of report has been set up, reports can be generated fairly quickly.

The procedure for creating reports using IQ software starts with the assumption that the Production Planner has logged in to the system.

Creating Reports Using IQ Software

NOTE: If using an X-Terminal, it may be necessary to add the following line to the **.Xdefaults** file in the home directory before performing the task for the first time:

```
iqx*background:                grey
```

NOTE: Commands in Steps 1 through 8 are typed at a UNIX system prompt.

1 Type **rlogin** ***hostname*** refers to the host (e.g., **e0mss21**, **g0mss21**, **l0mss21**, or **n0mss21**) on which GUIs are to be launched during the current operating session. Multiple hostnames can be specified on the same line.

2 Type **setenv DISPLAY** ***clientname*****:0.0** then press the **Return/Enter** key.

- Use either the X terminal/workstation IP address or the machine-name for the ***clientname***.
- When using secure shell, the DISPLAY variable is set just once, before logging in to remote hosts. If it were to be reset after logging in to a remote host, the security features would be compromised.

3 Open another UNIX (terminal) window.

4 Start the log-in to the Applications Server host by typing **/tools/bin/ssh** ***hostname*** (e.g., **e0mss21**, **g0mss21**, **l0mss21**, or **n0mss21**) in the new window then press the **Return/Enter** key.

- If you receive the message, **Host key not found from the list of known hosts. Are you sure you want to continue connecting (yes/no)?** type **yes** ("**y**" alone will not work).
- If you have previously set up a secure shell passphrase and executed **sshremote**, a prompt to **Enter passphrase for RSA key '<*user@localhost*>'** appears; continue with Step 5.
- If you have not previously set up a secure shell passphrase; go to Step 6.

5 If a prompt to **Enter passphrase for RSA key '<*user@localhost*>'** appears, type your ***Passphrase*** then press the **Return/Enter** key.

- Go to Step 7.

6 At the **<*user@remotehost*>'s password:** prompt type your ***Password*** then press the **Return/Enter** key.

7 Type **setenv ECS_HOME /usr/ecs/** then press the **Return/Enter** key.

- When logging in as a system user (e.g., cmshared), the ECS_HOME variable may be set automatically so it may not be necessary to perform this step.

8 Type **cd /usr/ecs/*MODE*/COTS/iqx5** then press **Return/Enter**.

- Change directory to the directory containing the IQ software (directory path may vary from site to site).
- The ***MODE*** will most likely be one of the following operating modes:
  - OPS (for normal operation).

– TS1 (for Science Software Integration and Test (SSI&T)).
– TS2 (new version checkout).

- Note that the separate subdirectories under /usr/ecs apply to (describe) different operating modes.

9 Type **iqx &** then press **Return/Enter**.

- If the GUIs are not displayed when the command **iqx** is given, try using **./iqx** instead.
- The **iqx IQView** GUI and either the **iqx Sybase Database Logon** GUI or **iqx Open IQView** GUI are displayed.
  – If IQViews have been defined previously, they are listed on the **iqx Open IQView** GUI; otherwise, a list of database tables is displayed.
- If the **iqx License** dialogue box is displayed, click on the **Cancel** button.
  – The **iqx IQView** GUI and either the **iqx Sybase Database Logon** GUI or **iqx Open IQView** GUI are displayed.

10 If the **iqx Sybase Database Logon** GUI is displayed, go to Step 16.

11 If the **iqx Open IQView** GUI is displayed and the desired IQView has been defined previously, perform Steps 12 through 14; otherwise, go to Step 15.

- If IQViews have been defined previously, they are listed on the **iqx Open IQView** GUI; otherwise, a list of database tables is displayed.

12 If the desired IQView has been defined previously, highlight the IQView to be opened by clicking on its entry in the list of IQViews.

13 Click on the **OK** button.

- The **iqx Sybase Database Logon** GUI is displayed.

14 Go to Step 16.

15 If the **iqx Open IQView** GUI is displayed and the desired IQView has not been defined previously, click on the **Add Database…** button.

  – The iqx **Sybase Database Logon** GUI is displayed.

16 When the **iqx Sybase Database Logon** GUI is displayed, type the appropriate entries in the following fields:

- **User:**
  – For example: pdpsUsr
  – The DAAC Database Administrator can provide the actual values to be entered.
- **Password:**
  – For example: **dbpa$$wd**

- **Database:**
  - For example: **pdps_TS1**
- **Server:**
  - For example: **x0pls02_srvr**
  - Click on the **OK** button.
- Either the **iqx Open IQView** GUI or the **iqx IQView** GUI is displayed.
  - If the **iqx Open IQView** GUI is displayed, continue with Step 19; if the **iqx IQView** GUI is displayed, go to Step 22.
  - Click on the **New IQView…** button.
- The **iqx Table Selection** GUI is displayed.
  - Move database table names between the **Available Tables:** and **Selected Tables:** lists as necessary by selecting (highlighting) the name of the table to be moved, then clicking on either the **Select** or **Remove** button (as applicable) to move the table name to the other list.
- Database tables and the columns within each table are described in the 311-series documents (e.g., 311-EMD-003, Release 7 Planning and Data Processing Subsystem Database Design and Schema Specifications for the EMD Project). The documents are available on the ECS Data Handling System (i.e., at http://edhs1.gsfc.nasa.gov).

21 When the desired table(s) has/have been moved to the **Selected Tables:** list, click on the **OK** button.
- The **iqx IQView** GUI is displayed.

22 Select **Output** → **Columnar** from the pull-down menu.
- The **iqx Columnar Output** GUI is displayed.

23 Move database table column names between the **Available columns:** and **Selected columns:** lists as necessary by selecting (highlighting) the column to be moved, then clicking on either the **Select** or **Remove** button (as applicable) to move the column name to the other list.
- The order in which columns are listed in the **Selected columns:** list is the order in which the columns will be listed in the eventual report.
- Database tables and the columns within each table are described in the 311-series documents (e.g., 311-EMD-003, Release 7 Planning and Data Processing Subsystem Database Design and Schema Specifications for the EMD Project). The documents are available on the ECS Data Handling System (i.e., at http://edhs1.gsfc.nasa.gov).

24 If changing the order in which columns are listed in the **Selected columns:** list, select (highlight) the column to be moved, then click on the **Up** or **Down** button as necessary to reposition the selected column.
- Highlighted column changes position in the **Selected columns:** list.

25 When the desired columns have been moved to the **Selected columns:** list, click on the **OK** button.

- The **iqx IQView** GUI is displayed.
- The columnar selections are listed on the **iqx IQView** GUI.

26 To generate a report make one of the following selections from the pull-down menu:

- **Execute → to Display –** to display the report on the terminal screen.
  - The iqx IQ Output GUI is displayed.
  - Go to Step 35 after viewing the report.
- **Execute → to Printer –** to print the report.
  - The iqx Execute to Printer GUI is displayed.
  - Go to Step 29.
- **Execute → to File –** to save the report in a file.
  - The iqx Execute to File GUI is displayed.
  - Continue with Step 27.

27 Type a valid ***path/filename*** in the **Name:** field of the **iqx Execute to File** GUI.

- For example: **/home/cmshared/reportfile**
  - Where **/home/cmshared/** represents the path and **reportfile** is the file name.

28 Click on the **OK** button.

- Go to Step 35.

29 Click on the **Printer…** button on the **iqx Execute to Printer** GUI.

- The **iqx Print Setup** GUI is displayed.

30 To list the available printers, first click on the option button associated with the **Printer** field.

- An option menu of printers is displayed.

31 Highlight the desired printer in the option menu.

- The desired printer is shown in the **Printer** field.
- For example: **Postscript printer one**.

32 If a report in landscape format is desired, click on the **Landscape** button.

33 Click on the **OK** button.

- The **iqx Print Setup** GUI is dismissed.
- The **iqx Execute to Printer** GUI is displayed

34 Click on the **OK** button.

35 To save the procedure/IQView, continue with Step 36; otherwise go to Step 43.

36 Select **File** → **Save Procedure As…** from the pull-down menu.

- The **iqx Save IQView** GUI is displayed.

37 Type a file name for the IQView in the name field.

38 Click on one of the following buttons if applicable:

- **Public**.
- Public read only.
- Private.

39 Click on the **OK** button.

- The **iqx Save Procedure** GUI is displayed.

40 Type a file name for the procedure in the name field.

41 Click on one of the following buttons if applicable:

- **Public**.
- **Public read only.**
- **Private.**

42 Click on the **OK** button.

43 Select **File** → **Exit** from the pull-down menu to exit from the **iqx IQView** GUI.

26.21.2 Formatting IQ Software Reports

IQ software report formatting can be achieve by exporting the report content to another application (e.g., a spreadsheet program) and using its features to structure the report.

26.22 Landsat 7 Error Handling Tool

The Landsat 7 Error Handling Tool provides the ECS Operations Staff with the ability to Merge/Demerge/Promote/Delete Landsat 7 granules using a command line interface. This tool works exclusively in the Science Data Server (SDSRV) database. The tool only modifies tables in the SDSRV database.

- The **Delete** command gives the user options to modify the DeleteFromArchive flag in the DsMdGranules table only or physically delete the granules from the archive and the inventory.
- The **Merge** command mimics the process done by the SDSRV and Landsat 7 Dynamic Link Library (DLL) during an ingest of Landsat 7 data.

- The **Demerge** operation allows the operations staff to separate incomplete combined granules and thus allows the recombination, Merging, of complete data sets.
- The **Promote** tool is used to associate granules with only a single format of data, bands 1-6 format 1/bands 6-8 format 2, with the appropriately combined subinterval and thus making the granule available for ordering. Before this tool, the data was unavailable.

26.22.1 Quick Start Using the Landsat 7 Error Handling Tool

Entering the following command starts the Landsat 7 Error Handling Tool:

EcDsSrDbL7ErrorHandling <mode> <Start Time> <Stop Time>

There are three command line parameters and they are used in combination with each other. Table 26.22.1-1 describes these parameters.

Table 26.22.1-1. Command Line Parameters of the Landsat 7 Error Handling Tool

Parameter Name	Description
Mode	Mode corresponding with the database to be modified
Start Time	Start time, insertTime, of the temporal range of granules to search. This will be used to search for unmerged subintervals/scenes when the Landsat 7 Error Handling Tool is started.
Stop Time	Stop time, insertTime, of the temporal range of granules to search. This will be used to search for unmerged subintervals/scenes when the Landsat 7 Error Handling Tool is started.

All of the parameters are mandatory. Before starting the tool the file "EcDsSrDbL7ErrorHandlingRC" should be modified to reflect the user's local environment:

```
# This file is used by EcDsSrDbL7ErrorHandling to setup environment
# variables, which makes EcDsSrDbL7ErrorHandling more tunable.
# MUST be updated to customize at sites.
#
export SYBASE=/tools/sybOCv11.1.1  # Directory where Sybase stuff resides
export SERVER=x0acgxx_srvr  # SQL server where ECS Science Data Server
# Database can be accessed
export ECS_HOME=/usr/ecs/
export SQSSERVER=x0acgxx_sqs322_srvr            # SQS server
export DBUSERNAME=EcDsScienceDataServer  # Valid user name to login into
# database
export DBPASSWD=xxxxxxx  # Password to access database
export DBNAME=EcDsScienceDataServerX  # Name of database
```

export WORKDIR=/usr/ecs/${MODE}/CUSTOM/data/DSS # Directory where the
script resides

export reportdir=/usr/ecs/${MODE}/CUSTOM/data/DSS   # Directory for report files
export tempdir=/usr/ecs/${MODE}/CUSTOM/data/DSS   # Directory for
temporary files
export
errorfile=/usr/ecs/${MODE}/CUSTOM/data/DSS/EcDsSrDbL7ErrorHandling.errlog
# File to use for holding any possible error messages

26.22.2 Landsat 7 Error Handling Tool Commands

The Landsat 7 Error Handling Tool provides the following granule modification options:

Initiate Merge of Landsat 7 Subintervals/Scenes from the SDSRV database. The selected Subintervals/Scenes must be passed to the tool via a file, which gets created during the start of the Landsat 7 Error Handling tool. The format of the input file is very specific, it is listed below.

Subinterval sample file input:

```
dbID    ShortName Insert Time        Path    Starting Row Ending Row       Assoc. File Name
10248   L70RF1    May 27 1998 9:26AM 172     44           50               SC:L70RF1.001:10248
10247   L70RF2    May 27 1998 9:26AM 172     44           50               SC:L70RF2.001:10247
10669   L70RF1    May 27 1999 9:25AM 172     44           50               SC:L70RF1.002:10669
10661   L70RF2    May 27 1999 9:25AM 172     44           50               SC:L70RF2.002:10661
```

Scene sample file input:

```
dbID    ShortName      Insert Time           Path    Row    Associated File Name
13530   L70RWRS1       May 27 1998 9:25AM    172     44     SC:L70RWRS1.001:13530
13531   L70RWRS2       May 27 1998 9:25AM    172     44     SC:L70RWRS2.001:13531
13532   L70RWRS1       May 27 1998 9:26AM    172     45     SC:L70RWRS1.001:13532
13533   L70RWRS2       May 27 1998 9:26AM    172     45     SC:L70RWRS2.001:13533
```

Promote Landsat 7 Subinterval/Scene from the SDSRV database. The selected Subintervals/Scenes must be passed to the tool via a file, which gets created when the operators starts the Landsat 7 Error Handling tool.

Demerge Landsat 7 Subinterval/Scene from the SDSRV database. The selected Subinterval/Scene must be passed to the tool by entering the geoid of the granule when prompted.

Delete Landsat 7 Subinterval/Scene from the SDSRV database. The selected Subintervals/Scenes must be passed to the tool by entering the geoid of the granule when prompted.

Generate list of Orphaned Landsat 7 Subintervals/Scenes in the SDSRV Database. The list of orphaned Subintervals/Scenes will be generated based on the start time and stop time parameters passed in as parameter #2 and #3 at invocation of the tool.

26.22.2.1 Initiate Merge of Landsat 7 Subintervals/Scenes from the SDSRV Database

This command has the form:

EcDsSrDbL7ErrorHandling <mode> <Start Time> <Stop Time>

This command invokes the tool. The script will display a menu of commands. To initiate a merge, the user should select option 3, “Merge Subintervals/Scenes”. The user will be prompted for a filename. The user should enter the name of the file that contains the format as described in section 26.22.2. The script will then return a failed or successful status and then return to the menu.

26.22.2.2 Promote Landsat 7 Subinterval/Scene from the SDSRV Database

This command has the form:

EcDsSrDbL7ErrorHandling <mode> <Start Time> <Stop Time>

This command allows the user to make data available that cannot be used. This command is used to make data with only one format appears as though both formats existed. The user enters the command and the script will display a list of options. The user should select option 5,”Promote Orphaned granules”. The user will be prompted for a filename that contains the format as described in section 26.22.2. After entering the filename, the script will return a failed or successful status and then return to the menu.

26.22.2.3 Demerge Landsat 7 Subinterval/Scene from the SDSRV Database

This command has the form:

EcDsSrDbL7ErrorHandling <mode> <Start Time> <Stop Time>

This command is used for incorrectly combined data sets. The user should select option 4, “Demerge L70RF1/F2 granules”, from the menu. The script will prompt the user for the geoid of the granule to demerge. An example of a geoid is SC:L70R.001:12345. The first part is the type of the granule. SC represents science granules. The second part is the subtype and version of the granule. The last part is the dbId of the granule. This uniquely identifies the granule in the Science Data Server’s database.

26.22.2.4 Delete Landsat 7 Subinterval/Scene from the SDSRV Database

This command has the form:

EcDsSrDbL7ErrorHandling <mode> <Start Time> <Stop Time>

After entering the above command, the user will see a list of options. The user will should select option 6, ”Delete Unmerged granules.” The script prompts the user for the geoid of the granule to delete. The user will be prompted for confirmation of the deletion. The user also will be prompted to determine if the granule should be deleted from the archive and the inventory.

26.22.2.5 Generate List of Orphaned Landsat 7 Subintervals/Scenes in the SDSRV Database

This command has the form:

EcDsSrDbL7ErrorHandling <mode> <Start Time> <Stop Time>

The first time the script is invoked it always searches the SDSRV database for orphaned granules within the time range given by the start time and stop time input parameters. After that, the user can generate the list by selecting command #2, "Update files on /tmp directory". This will create two files in the /usr/ecs/<MODE>/CUSTOM/data/DSS directory. One of the files contains all the unmerged subintervals, the file name is unmergedsubintervals. The other file contains all the unmerged scenes, the file name is unmergedscenes. The user may use these files for the merged Landsat 7 granules and promoted Landsat 7 granule options.

26.22.3 Required Operating Environment

For information on the operating environment, tunable parameters, and environment variables refer to 910-TDA-022, Custom Code Configuration Parameters for ECS.

26.22.3.1 Interfaces and Data Types

Table 26.22.3.1-1 lists the supporting products that this tool depends upon in order to function properly.

Table 26.22.3.1-1. SDSRV Database Interface Protocols

Product Dependency	Protocols Used	Comments
SDSRV Database	SQL	via SQL server machine

26.22.4 Databases

The Landsat 7 Error Handling tool does not include the direct managing of any database. It has an interface with the Science Data Server Database: however this interface is based on a simple parameter passing function. For further information of the Science Data Server Database refer to 311-EMD-004, Release 7 Science Data Server Database Design and Schema Specifications for the EMD Project.

26.22.4.1 Special Constraints

The Landsat 7 Error Handling Tool doesn't require any servers to be running. It is strongly recommended that as little as possible should be going on in the SDSRV database while the tool is being used.

26.23 Deleting Granules

The system provides a **Granule Deletion** tool, complementing the automatic, scheduled deletion capability that permits operators to delete products produced and archived by the Planning and Data Processing subsystems on a scheduled basis (e.g., deletion at a certain time (configurable by the operator) after product creation.

The **Granule Deletion** tool allows operators to delete products on demand. There are a variety of circumstances that may require deletion on demand, such as:

- New PGE versions have been created and are used to reprocess large amounts of past data, creating new ESDT versions. As reprocessing progresses, operations deletes the granules for the old ESDT versions from the archive and inventory.
- It is determined that certain lower-level (e.g., Level 2) products are of little or no interest to the science or public user community. In concert with the science teams, DAAC operations personnel decide to remove these products from the inventory. Since the products are still referenced by higher-level products as inputs, the DAAC decides to keep the inventory records for production history purposes.
- One or more granules were found defective and were reprocessed on an individual basis. When the reprocessing is complete, the operator wishes to delete the old, defective granule(s) from the inventory.
- A DAAC has extended ECS with subsetting services. The subsetted products are produced outside ECS, but are then inserted into the ECS archive to take advantage of the ECS distribution capability. The DAAC writes a script to delete the subsetted products on a regular basis.

26.23.1 Deletion Capability and Features

The Science Data Server has provided an application programming interface (API) for deleting granules from the archive, or from both the archive and inventory since earlier releases, but the Granule Deletion tool adds a front-end command-line utility that provides several ways for selecting granules for deletion. Confirmation is generally required so that granules are not inadvertently deleted. However, the confirmation may be suppressed so that operators can run regularly scheduled deletion scripts using background execution. This suppression possibility presents an opportunity for inadvertent loss of data and so must be used with care and only after thorough testing of any deletion script.

The Science Data Server captures deletions and related errors in the application log. Operators may also specify a separate and independent delete log for immediate analysis of the success or failure of a delete operation.

26.23.1.1 Deletion Sequence

The deletion of granules from the archive involves three elements, and therefore actually occurs in stages. Two of the elements are scripts that address the Science Data Server (SDSRV), and the third is a part of the Storage Management (STMGT) Graphical User Interface (GUI).

- For the first stage, a delete script applies deletion checks to the selected granules, "logically" deleting from the inventory those granules that satisfy the checks. These granules are flagged as 'deleted' and can no longer be accessed, but their inventory entries are not yet removed. The deletion flag consists of a time stamp recording the logical deletion time.
- The second stage is actual deletion from the inventory, which occurs when the operations staff runs the physical deletion script. The script removes all inventory rows for granules that were flagged as 'deleted,' and produces the list of the granule files that are now eligible for deletion from the archive. That list is transferred to the STMGT database. The operations staff controls the lag time between logical deletion and physical deletion. That lag time is entered into the physical deletion script, which deletes only inventory entries for granules that have been logically deleted prior to that time period.
- STMGT provides a GUI screen that allows the operator to initiate the removal from the archive of the files listed its deletion table (populated by SDSRV). STMGT creates requests to the archive servers to delete files. The STMGT GUI can be used to look at the state of the deletion requests. Files that are successfully deleted have their associated rows removed from the STMGT database table.

Periodically, as sufficient data removal from the archive makes it appropriate, operations may elect to reclaim the tape space and recycle archive tapes. The AMASS software commands (*volcomp*, *volclean*, *volformat*, *volstat*) are used for that purpose.

26.23.2 Granule Deletion

This process of Granule Deletion is for operators to delete granules from the inventory/archive on demand. Deletion of Granules is a three-part process.

- Granule Deletion Client
- Deletion Cleanup Utility
- STMGT GUI

1. First, a command line SDSRV utility, called the **Granule Deletion Client**, is available for selecting granules to be marked for deletion from the archive (DFA), with an option to also mark for delete from the inventory (PHYSICAL DELETE). A confirmation is generally available, but can be suppressed to allow for background execution. The Science Data Server logs will be inspected for messages related to deletions, and a separate log is available for immediate inspection of only deletion messages.

2. The second, a SDSRV script, called the **Deletion Cleanup Utility**, is run to send those granules that have been marked for deletion to the STMGT database so they can be physically deleted from the archive. In addition, if granules were marked for Physical Delete in SDSRV, they will also be deleted from the SDSRV database. A lag time is used so the operator can request that not all the granules marked for deletion are immediately deleted. Granules will be deleted when the Deletion Cleanup Utility is executed when they have been marked for deletion before <today - lagtime>. This lag time can be overridden by entering a lag time of 0, in which case all granules marked for deletion will be immediately deleted.

3. The third and last part of the process is the **STMGT GUI** can then be used to complete deletion of data granules from the archive (AMASS). Granules are selected for deletion from the STMGT GUI via the datatype name (ESDT ShortName) and Version ID.

4. Using the Granule Deletion Client, granules can be selected for deletion several different ways:
   - ESDT ShortName, Version and granule time coverage
   - ESDT ShortName, Version and granule insert time range
   - Separate Input file containing SDSRV Granule IDs
   - Separate Input file containing ShortName, Version, and Local Granule ID (Logical Granule ID noted in the ticket is referred to in this test as Local Granule ID, which is the name of the parameter as it resides in the database)

5. The number of granules returned can be verified by making the same query in the database using SQL commands. Granules can be deleted from both the inventory and the archive, from the archive only, in the foreground, and via background (cron) jobs. It can be verified that all occurrences of a file in the archive are deleted when requested, such as from a primary and a backup archive location. It can be verified that Browse, QA and PH granules associated with physically deleted granules are also deleted if not referenced by any other granules.

   Error conditions include:
   - Attempts to delete granules that are still being referenced by other granules
   - Unauthorized users are indeed halted from deleting data
   - Recovery from Archive Server, Science Data Server and Science Data Server DBMS server faults occurring during deletion requests

6. Using a user ID authorized for granule deletions, the operator has a number of interfaces to consider for the deletion of granules, including the mechanisms for selecting the granules to be deleted and the confirmation of the deletion.
   - Granules can be selected by ESDT short name, ESDT version, and granule time coverage
   - Granules can be selected by ESDT short name, ESDT version, and granule insert time range
   - Granules can be specified in a separate input file containing either SDRV Granule Ids or Logical Granule Ids
   - The input file can list granules belonging to collections belonging to different logical volume groups.
   - The operator can optionally list the geoID and logical granule ID of each of the granules selected for deletion.
   - The number of granules selected for deletion is displayed to the operator and the operator is asked to confirm the deletion
   - The operator can suppress the confirmation prompt via a command line argument
   - Granules are tagged for deletion from inventory and archive or from archive only, depending on operator choice

- By default, BROWSE, QA, and PH granules associated with physically deleted granules are deleted if no longer referenced otherwise
- The operator can suppress deletion of BROWSE, QA and PH granules
- The files associated with the deleted granule are deleted from all archive locations (including back-up locations)
- The deletions are logged as required to the SDSRV application log file and the operator specified granule deletion log file
- Granule deletions can be performed during normal SDSRV processing, i.e., while other requests such as insert, acquire, and searching are in progress

26.24 Additional Information on the Preparation of Earth Science Data Types (ESDTs)

Every science data product generated and archived by the ECS must be described to the system by metadata that are put into an inventory and then used to retrieve and distribute the data to users of the system. The ECS Earth Science Data Model organizes the metadata into groups of related attributes and services to be performed on the data products. Granules of the same type of science data are grouped into collections. Every collection is described by an Earth Science Data Type (ESDT) and is made known to the system by an ESDT descriptor file and associated software code that is built into the Data Server's dynamic link library (DLL) to perform the services. The ESDT descriptor is composed of sections containing the following information:

- Collection level metadata attributes with values contained in the descriptor.
- Granule level metadata attributes whose values are supplied primarily by the Product Generation Executives (PGEs) during runtime.
- Valid values and ranges for the attributes.
- List of services for the data and events that trigger responses throughout the system.

The ESDTs for all data collections to be input to or output from the PGEs must be built and registered into the ECS before any of the PGEs can be run under the automated processing system.

ECS has collected information from the Instrument Teams on the ESDTs needed for their science software in Release 6B. This information has been baselined and a set of ESDT descriptor files have been built and tested according to this baseline. The baselined ESDT descriptor files reside in Release 6B under ECS configuration management in a VOB called the ECS Configuration Area. Since science software has been developed to the baseline, any changes to the baselined ESDT descriptor files should be rare.

26.24.1 Comparing Granule Level Metadata

A PGE accesses granule level metadata attributes and values via a Metadata Configuration File (MCF). There is typically one MCF for each output data set. The ESDTs that have been built and registered in the ECS contain a section for granule level metadata attributes and values for each data set. In terms of content, the MCFs and the granule level metadata section of the corresponding ESDT descriptor files have to be in sync.

Few changes are expected in the Inventory section of the MCF. Changes are more likely to be expected in the Archive section of the MCF and in the Product Specific Metadata. If there are any changes, a new version of the baselined ESDT descriptor file must be generated.

26.24.2 Installing/Removing ESDT/DLL Using SDSRV Operator GUI

Before the ECS can process data, an Earth Science Data Type must be installed into the system via the Science Data Server (SDSRV). The ESDT allows the system to recognize a particular data type and also provides services for accessing the data in the form of a Dynamic Link Library (DLL). The following procedures give step-by-step instructions on configuring the ESDT and installing the ESDT using the Science Data Server GUI.

26.24.2.1 Installing a Single ESDT with Dynamic Link Library (DLL)

1 Copy the ESDT descriptor file and ESDT/DLL file from the source directory to the directory under the current mode of operations, The ESDT descriptor files are installed in the specified mode.

- DLL's located : /usr/ecs/<mode>/CUSTOM/lib/ESS
- ESDT Descriptors Located: /usr/ecs/<mode>/CUSTOM/data/ESS

2 Ensure that the following servers are currently executing: **Data Dictionary Service** on the appropriate DMGHW HWCI server machine, **Science Data Server** on the appropriate ACMHW HWCI server machine and the **Subscription Service** that operates on the appropriate CSS server machine.

3 Start the **SDSRV GUI** by entering the following at the UNIX prompt on the SDSRV GUI workstation:

- On workstation **x0acs##,** at the UNIX prompt in a terminal window, type as in step a, below, then enter your user id and password.
- NOTE: The x in the workstation name will be a letter designating your site:
- g = GSFC, m = SMC, l = LaRC, e = EDC, n = NSIDC, o = ORNL, a = ASF, j = JPL, p = PVC; the ## will be an identifying two-digit number (e.g.,g0acs03 indicates a Science Data Server Subsystem (SDSRV) workstation at GSFC).
- Then rlogin, and enter **setenv DISPLAY <local_workstation IP address>:0.0**. The **<ipaddress>** is the ip address of **x0acs##**, and xterm is required when entering this command on a Sun terminal.

26.24.3 Quick Start Using the Science Data Server

To invoke the ECS Science Data Server GUI, the user types the following command line:

1 **telnet** to (SDSRV) **p0acs03** [e.g.]

2 login: **ID, password:**

3 **cd /usr/ecs/<mode>/CUSTOM/utilities/EcDsSdSrvGuiStart <mode> Enter: >EcDsSdSrvGuiStart <mode>**

- <**mode**> is the ECS Mode for the execution, e.g., **OPS, TS1.**
- This command will bring up the GUI with the main screen appearing. The user can then initiate the actions described in the following section.

4 On the main screen select the **Data Types** tab. A list of the ESDTs that have already been installed on the SDSRV will be displayed.

5 Click the **Add** button, this will bring up a first smaller GUI. Go below to bring up the **Add Data Type** window, this will bring up a second small GUI.

6 **Descriptor Filename:** enter path to where the ESDT/DLL is located, including the full ESDT descriptor. Click **OK**. Control goes back to the first GUI.

- **Note:** Within the ESDT descriptor there is a DLL Filename identifying which DLL is to be used with this ESDT.
- The descriptor filename and DLL Filename will require the complete directory path name as part of the file name which is the same directory as was specified in step 1 above. (isolate the particular Data Type from the larger List, by using a unique sequence of letters or numbers at the end of the full path to better identify the Data Types i.e., **/*___*).**
- To specify specific directories, the File button to the right of the Descriptor Filename and DLL Filename data entry fields will bring up a standard file selection GUI for this purpose. Also note that the Archive ID field will be constructed using the DSS Storage Management Staging Server UR that is found in the Science Data Server configuration file. The Science Data Server Configuration file is located in: **/usr/ecs/<mode>/CUSTOM/cfg/EcDsScienceDataServer.CFG.**
  Example: If the **DSSSTMGSTAGEINGSERVERUR** field was set to **DRP1_OPS:VG1** then the Archive ID fields will automatically be set to **DRP1_OPS.**

7 Click the **Ok** button, this will cause the **Add Data Type** window to initiate installation of the ESDT/DLL into the Science Data Server.

8 The Science Data Server GUI will respond in a short time with a window stating that: **MM/DD/YY HH/MM Finished adding ESDTs**. Also, the ESDT will appear alphabetically on the **Science Data Server - Data Types** list under the **Data Types** tab.

26.24.3.1 Validating Successful ESDT Installation

Criteria for success:

- The **SDSRV** will display an Event ID to the fact that a new ESDT has been installed successfully.

- The following servers will also need to have acknowledged a successful ESDT Event ID before additional work can be done: **DDICT, SBSRV**.

26.24.4 Science Data Server

The SDSRV provides the SDPS with a catalog of Earth Science Data holdings, and the Earth Science Data Type services that operate on the data. The SDSRV provides a catalog of metadata describing the archived data holdings of the SDPS and provides mechanisms to acquire the data from the archive. The SDSRV also provides data type service on the catalog and a data reduction or sub-setting and reformatting service.

The Science Data Server Operator GUI provides the operator two major functions, the management of Earth Science Data Types and the management of all types of requests the Science Data Server Operator is involved with. Further details on these two functions are given in Table 26.24.4-1.

Table 26.24.4-1. Common ECS Operator Functions Performed with the Science Data Server Operator GUI

Operating Function	GUI	Description	When and Why to Use
Manage Science Data Server Earth Science Data Types (ESDTs)	Data Types Tab	Allows operators to manage the ESDTs offered by the Science Data Server	As needed, to manage data type descriptor information and add and update ESDTs
Manage Data Server System Requests	System Requests Tab	Allows operators to manage all the requests within each data server component	As required, to manage requests in each data server component

26.24.4.1 Science Data Server Main Screen

The ECS Science Data Server operator GUI has two tabs that provide access to each one of the components' screens.

- The Earth Science Data Type Manager is accessed through the **Data Types** tab
- The System Request Manager is accessed through the **System Request** tab.

The operator can select from the menu bar items at the top of the Science Data Server Operator window for getting help and activating less-frequently used functions. The menu bar capability is available on all Science Data Server Operator GUI screens. The following menus are available:

- **File** - which includes the following item:
  - **Exit** (Ctrl-Q) - Exit application (graceful exit).
- **View** - functionality has not been determined as of this time (TBS).

- **Options** - This menu includes the *System Settings* item that opens the Server Polling Options window. Polling of the data server can be switched On/Off and the SdSrv Polling rate can be adjusted through this window.
- **Help** - which provides context sensitive help.

Table 26.24.4-2 describes the information fields on the Server Polling Options screen.

Table 26.24.4-2. Science Data Server - Server Polling Field Description

Field Name	Data Type	Size	Entry	Description
Polling Rate	integer	4 digits	Optional	Specify the rate at which the Science Data Server Operator GUI is updated with data coming from the Data Server. The polling rate default is 120 seconds.

26.24.5 Data Types Tab

The Data Types Tab is the default screen of the Science Data Server Operator GUI. This window provides operations personnel at the DAAC the capability to view descriptor information, add new ESDTs and update ESDTs. A list of currently installed ESDTs is shown along with a version number and a brief description of the structure for an ESDT. Additional information that describes the structure, contents, and services that each existing ESDT provides can be viewed by selecting the data type and clicking on the *View* button.

Table 26.24-5-1. describes the Science Data Server Operator - Data Types fields.

Table 26.24.5-1. Science Data Server Operator - Data Types Field Description

Field Name	Data Type	Size	Entry	Description
Data Type ID	character	8	System generated	Uniquely identifies the specific type of ESDT.
Name	Character	25	System generated	Name of ESDT.
Version	Integer	3	System generated	Version number of ESDT, assigned starting at 1.
Description	Character	255	System generated	Includes structure and services available for an ESDT.
Find	Character	255	Optional	This functionality is provided in order to help the user browsing very long ESDT lists.

In addition, the following buttons are provided:

- **View** displays ESDT descriptor information (read-only) and its associated dynamic data link library (DLL) filename. Descriptor information consists of groups, objects, and keywords that define an ESDT's metadata, advertised services, subscribable events, data dictionary information, validation criteria, and science parameters. Descriptor information is necessary for the Science Data Server to properly configure itself to perform services related to an ESDT. A DLL is an executable library that is loaded dynamically when needed to fulfill ESDT services.

- **Close** exits the dialog without performing any operations.
- **Help** displays on-line help information.
- **Add** opens the Data Type Dialog, which is used to add a new ESDT to the existing installed list of data types based upon input information. The SDSRV GUI has the capability to install multiple ESDTs. Click on the **File…** button to display a list of descriptor filenames to choose from instead of typing them in. Multiple descriptor files can be selected. Click the **OK** button to add the data type. If no error messages appear, then the operation has been successfully completed. Click the **Cancel** button to close the dialog without performing an operation. Click the **Clear** button to start all over again the process of filling in new information. Click the **Help** button to display on-line help information.

Table 26.24.5-2, describes the Science Data Server - Add Data Type fields.

Table 26.24.5-2. Science Data Server - Add Data Type Field Description

Field Name	Data Type	Size	Entry	Description
Descriptor Filename	Character string	25	required	Name of an ASCII file containing the ESDT descriptor file.

- **ADD ESDT** GUI is similar in use to the Update ESDT GUI and guideline.
- **Update** opens the Update ESDT Dialog, which is used to update an ESDT to the installed list of data types based upon input information. The SDSRV GUI provides the capability to update multiple ESDTs at one time. The Science Data Server needs to be running in **Maintenance mode** to accept this operation. Click on the **File…** button to display a list of descriptor filenames to choose from instead of typing them in. Multiple descriptor files can be selected. Click the **OK** button to update the data type. If no error messages appear, then the operation has been successfully completed. Click the **Cancel** button to close the dialog without performing an operation. Click the **Clear** button to start all over again the process of filling in new information. Click the **Help** button to display on-line help information.

Table 26.24.5-3, describes the Science Data Server - Update Data Type fields.

Table 26.24.5-3. Science Data Server - Update Data Type Field Description

Field Name	Data Type	Size	Entry	Description
Descriptor Filename	character string	255	required	Name of an ASCII file containing the ESDT descriptor file.

- **Refresh/Reconnect** updates the data type information screen with current information.
- **Operator Messages** provides the functionality that displays informational and error messages to the user.

26.24.6 System Requests Tab

Clicking the System Requests tab will bring up the System Management Requests window. This window provides operations personnel at the DAAC the capability to monitor requests the Science Data Server is working with. All requests within the Science Data Server are displayed. The columns of the list can be sorted by positioning the cursor and clicking on the appropriate column of interest. The requests can be filtered by positioning the cursor and clicking on the **Filter** button and entering the attributes on which to filter.

Table 26.24.6-1, describes the System Management Requests Window fields.

Table 26.24.6-1. System Management Requests Field Description

Field Name	Data Type	Size	Entry	Description
Request ID	character	255	system generated	Unique identifier for the request.
Requester	variable character	100	system generated	Identifies the user that submitted the request.
Service Request	character	25	system generated	Types of requests handled are Insert, Acquire, and Delete
Status	character	20	system generated	Possible states are Submitted, Queued, Executing, Failed_Retryable, Failed_Fatal, Failed_Unknown and Done
Priority	variable character	20	system generated	Priority of the data server system requests, i.e., Express, Very High, High, Normal(default), Low.
Find	character	255	optional	If the list is too long, this field can be used to search for an entry

In addition, the following buttons are provided:

- **Change Priority:** changes the priority of each selected request through a pull-down menu. Possible values are: Express, Very High, High, Normal (default) and Low.
- **Apply** allows the operator to commit to the priority change selected through the change priority button.
- **Filter…** brings up the System Management Filter Requests dialog that provides a selection of attributes on which to filter for the list of System-wide requests. Filter on system management requests by entering the desired information, then clicking on the Request ID or Requester radio button for the desired attribute. Return to the original list of requests by clicking on the All Requests radio button. Click on other filters associated with State and Priority by clicking on the toggle button. Filter on every attribute associated with a category by clicking the **All** button or clear a category of filters by clicking on the **None** button.
- **Memory State:** it monitors the current memory state of the data server in regards to values that are set on the server side through configuration parameters. Possible values are: Normal(green color), Low(yellow), Very Low(red). This functionality will only be visible if the server's DSSMEMORYMONITORDISABLEFLAG is off.

Table 26.24.6-2 describes the System Management Filter Requests Dialog fields.

Table 26.24.6-2. System Management Filter Requests Field Description

Field Name	Data Type	Size	Entry	Description
Request ID	character	255	system generated	Unique identifier for the request.
Requester	variable character	100	system generated	Identifies the user that submitted the request.

In addition, the following buttons are provided:

- **OK** implements filter criteria, and the dialog closes.
- **Apply** implements filter criteria, and the dialog remains open for additional filtering.
- **Cancel** closes the dialog without saving
- **Help** displays on-line help information.
- Back to the System Requests tab description, **Operator Messages** provides informational and error messages to the user.
- **Refresh** causes the Data Server to be polled for an update on Requests.

26.24.7 Required Operating Environment

For information on the operating environment, tunable parameters, and environment variables refer to 910-TDA-022, Custom Code Configuration Parameters for ECS.

26.24.7.1 Interfaces and Data Types

Table 26.24.7.1-1, lists the supporting products that this tool depends upon in order to function properly.

Table 26.24.7.1-1. SDSRV GUI Interface Protocols

Product Dependency	Protocols Used	Comments
SDSRV GUIs	X-11	via client libraries

26.24.7.2 Databases

The Science Data Server Operator GUI does not include the direct managing of any database. It has an interface with the Science Data Server Data Base: however this interface is based on a simple parameter passing function. For further information of the Science Data Server Data Base refer to 311-EMD-004, Release 7 Science Data Server Database Design and Schema Specifications for the EMD Project.

26.24.7.3 Special Constraints

The Science Data Server Operator GUI runs only if the Science Data Server is running in the background. Note also that at the moment the Science Data Server GUI is started through a command line that specifies the configuration file that is used to initialize the GUI Application.

26.24.7.4 Outputs

There is no processing associated with the operation of this GUI. The information provided to the operator is retrieved from the Data Server Database described in the previous sections and displayed through the screens discussed therein.

26.24.7.5 Event and Error Messages

Both event and error messages are listed in Appendix A of document 609-EMD-001, Operations Tools Manual for the EMD Project.

26.24.8 Browser to View ECS SDSRV Database

1 Connect to the **SDSRV (p0acs03)** database with login information as follows:

- Server Name: **p0acg01_srvr**
- User Name: **sdsrv_role**
- Password: **welcome**
- The Browser lets you view all the tables in the **SDSRV** database with the mode you have selected. Example**: <EcDsScienceDataserver1_TS1**> for the PVC.

26.24.8.1 Using ISQL Commands in the SDSRV

1 **ssh p0acs03**

2 **Password:**

3 **isql -Usdsrv_role -Pwelcome -p0acg01_srvr**

4 **1> use EcDsScienceDataserver1**

5 **2>go**

6 **3>select * from <choose table_name>**

7 **4>go**

26.24.8.2 Wisqlite is an ISQL Browser for Viewing ECS Databases

1 login: **ssh p0pls02**

2 **cd /tools/tcl/bin**

3 Execute: **wisqlite &**

4 User ID: **pdps_role**

5 PW: **<password>**

6 Select server_name by clicking on Server button p0pls01_srvr

7 Select: **Sign on**

8 Enter: pdps, objects, tables

9 **Select * from <table your choice>**

26.24.8.3 Tables to Track .met files

The following tables are useful in tracking down problems in insert ***.met**:

- DsDeDictionaryAttribute
- DsMdAddtionalAttributes
- DsMdCollections - View with ECS Assistant only
- DsMdGranules - View with ECS Assistant only

26.24.9 Removing ESDTs from Archive Area Using Command Line Scripts

26.24.9.1 Removing ESDTs Using Command Line Script Procedures

1 **telnet** to (SDSRV) **p0acs03**[e.g.]

2 **login:, password**: <log into cmshared>

3 **cd /usr/ecs/TS1/CUSTOM/local**

4 Run script: **cleanesdt.csh <descriptor_file_name>**
- The ESDT will be removed from all three servers, SDSRV, SUBSRV, & DDICT. Check SDSRV GUI to see if removal was successful.

5 Now you have to log into each server and bring each of the four servers mentioned above, down by using the **KILL Script** and the **START Script** that have been specifically tailored for each server at the path identified below in each subsystem:

cd/home/cmshared/bin/

- SDSRV you are already on, p0acs03, SBSRV is on p0ins01, DDICT is on p0ins02.

26.24.9.2 Staging ESDTs for Installation

1 Copy into a directory from which the Science Data Server will use to bring ESDTs into the Science Data Server Data Base

2 Once you have the descriptor file copied into <your_directory> in the **p0ins01** machine, you have to login into **p0acs03** and copy the file into **/usr/ecs/TS1/CUSTOM/data/ESS** directory. To do this, log into **p0acs03.** From the directory **/usr/ecs/TS1/CUSTOM/data/ESS,** Type command,

cp /home/your_directory/ESDTs/filename

(Where /home/your_directory/ESDTs is the directory into which you stored the descriptor file.)

Now you have the copy of the latest descriptor file in the **ESS** directory.

26.24.10 Installing ESDTs Using the Science Data Server GUI

Procedures:

1 **telnet** to (SDSRV) **p0acs03**[e.g.]

2 **login:, password**: <always log into **cmshared**> to enable use of isql commands and to bring servers up and down.

3 **cd /usr/ecs/TS1/CUSTOM/utilities**

4 If an ESDT has to be removed before replacing it, run script: **cleanesdt.csh <descriptor_file_name>**

- The ESDT will be removed from all three servers, SDSRV, SUBSRV, & DDICT. Check SDSRV GUI to see if removal was successful.

5 Now you have to log into each server and bring each of the four servers mentioned above, **down** by using the **KILL Script** and the **START Script** that have been specifically tailored for each server at the path identified below in each subsystem:

cd/home/cmshared/bin/

- SDSRV is on, p0acs03, SBSRV is on p0ins01, and DDICT is on p0ins02.

6 Once you know that all four Science Data Servers are down, then log into **p0acs03** machine, go to directory **/usr/ecs/TS1/CUSTOM/utilities** and execute command

EcDsScienceDataServerStart TS1 StartTemperature cold

- This script will bring up the four Data Servers.
- Then check the status of the servers by using a Server Monitor GUI WHAZZUP. Once all the servers are up, then you can start installing the ESDT.

7 From **p0acs 03,** go to the directory, **/usr/ecs/TS1/CUSTOM/utilities,** type command:

EcDsSdSrvGuiStart TS1

8 The Science Data Server GUI will be brought up. You will see the list of already installed ESDTs there. To add new one, click on the **Add** button. You will get the **Add ESDT** window.

- This window is pointing to: **/usr/ecs/TS1/CUSTOM/data/ESS** where ESDTs have been staged.

9 Click on the **File** button to select the descriptor file. Select the correct .desc file and click on **OK**. Enter ArchiveID as **DRP1_TS1.** Click on **OK**. Wait and check the message at the bottom of the window. You will see message – **Successfully added**…Click the **Refresh** button and you will see recently added ESDT into the list.

26.24.10.1 Re-install or Update ESDTs Using the Science Data Server GUI

Only the Science Data Server (SDSRV) needs to be brought down/up using the Kill and Start scripts before doing step 4 below.

Procedures:

NOTE: If SDSRV is already running, you will have to Kill it first and then bring it up in the **maintenance mode**.

1 **telnet** to (SDSRV) **p0acs03**[e.g.]

2 **login:, password**:

3 **cd /usr/ecs/TS1/CUSTOM/utilities**

- Log into the (SDSRV) using **Maintenance Mode.** This locks out other users while the data base is being update with ESDTs.

4 Execute command :

EcDsScienceDataServerStart TS1 StartTemperature maintenance

- This script will bring up the Science Data Server.
- Then check the status of the SDSRV using a Server Monitor GUI **WHAZZUP.** Once all the servers are up, then you can start installing the ESDT.

5 From **p0acs 03,** go to the directory, **/usr/ecs/TS1/CUSTOM/utilities,** type command:

EcDsSdSrvGuiStart TS1

6 The Science Data Server GUI will be brought up. You will see the list of already installed ESDTs there. To **Update** an ESDT, click on the Update button. You will get the **Update ESDT** window.

- This window is pointing to: **/usr/ecs/TS1/CUSTOM/data/ESS** where ESDTs have been staged.

7 Click on the **File** button to select the descriptor file. Select the correct .desc file and click on **OK**. Click on **OK**. Wait and check the message at the bottom of the window. You will see message **– Successfully added**…Click the **Refresh** button and you will see recently added ESDT into the list.